Package com.selligent.sdk

Interface Summary

Interface	Description
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
SMInAppMessageReturn	Interface used to implement the code that will be executed after retrieving the In-App contents
SMNotificationCallback	Deprecated *Since 1.3, a broadcast is performed at the click on a button (cf.*

Class Summary

Class	Description
SMApplication	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 in the foreground for the push from the Selligent Mobile Platform.
SMInAppContent	An In App content
SMInAppContentHtmlFragment	This class implements a fragment that will display one or several HTML contents.
SMInAppContentImageFragment	This class implements a fragment that will display In App Content containing an image.
SMInAppContentUrlFragment	This class implements a fragment that will display In App Content containing an url.
SMInAppMessage	An In App message
SMLink	A link of an `SMInAppContent`.
SMManager	Singleton object used to interact with the Selligent Mobile SDK.
SMMapMarker	This class represents a point on a map.
SMNotificationButton	Object containing all the data used to implement a button in a notification/message.
SMObserverManager	This class is used to observe some events of the SDK.
SMSettings	Configuration object passed to the 'start' method of SMManager.

Enum Summary

Enum	Description
SMClearCache	
SMContentType	Enum listing the different types of In App Content
SMInAppContent.DisplayMode	Enum listing the different display mode.
SMInAppRefreshType	Enum with the different values for the refresh of the In App messages and In Ap contents.

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.
SMThemeCategories	Deprecated

com.selligent.sdk

Class SMApplication

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

All Implemented Interfaces:

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

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

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

Since:

`1.0`

Version:

`3.5`

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, 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, 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_SERVICE, DOWNLOAD_SERVICE, DROPBOX_SERVICE, EUICC_SERVICE, FILE_INTEGRITY_SERVICE, FINGERPRINT_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_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, 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_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 and Description
SMApplication()

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method and Description
void	**onCreate**()

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

attachBaseContext, bindIsolatedService, bindService, bindService, bindServiceAsUser,

checkCallingOrSelfPermission, checkCallingOrSelfUriPermission, checkCallingPermission, checkCallingUriPermission, checkPermission, checkSelfPermission, checkUriPermission, checkUriPermission, clearWallpaper, createAttributionContext, createConfigurationContext, createContextForSplit, createDeviceProtectedStorageContext, createDisplayContext, createPackageContext, createWindowContext, databaseList, deleteDatabase, deleteFile, deleteSharedPreferences, enforceCallingOrSelfPermission, enforceCallingOrSelfUriPermission, enforceCallingPermission, enforceCallingUriPermission, enforcePermission, enforceUriPermission, enforceUriPermission, fileList, getApplicationContext, getApplicationInfo, getAssets, 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, getResources, getSharedPreferences, getSystemService, getSystemServiceName, getTheme, getWallpaper, getWallpaperDesiredMinimumHeight, getWallpaperDesiredMinimumWidth, grantUriPermission, isDeviceProtectedStorage, isRestricted, 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, 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

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

Constructor Detail

SMApplication

```
public SMApplication()
```

Method Detail

onCreate

```
public void onCreate()
```

Overrides:

```
onCreate in class android.app.Application
```

com.selligent.sdk

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:

`3.5`

See Also:

`SMEvent`, `SMManager.sendEvent(SMEvent)`

Method Summary

All Methods | Instance Methods | Abstract Methods

Modifier and Type	Method and 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

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:

`3.5`

Enum Constant Summary

Enum Constants

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

Method Summary

All Methods | **Static Methods** | **Concrete Methods**

Modifier and Type	Method and Description

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

Methods inherited from class java.lang.Enum

clone, compareTo, equals, finalize, 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

com.selligent.sdk

Enum SMContentType

java.lang.Object
    java.lang.Enum<SMContentType>
        com.selligent.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:

`3.5`

Enum Constant Summary

Enum Constants

Enum Constant and Description
Html
Image
Url

Method Summary

All Methods | **Static Methods** | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method and 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

clone, compareTo, equals, finalize, 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)
```

com.selligent.sdk

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:

3.5

See Also:

SMManager.sendDeviceInfos(SMDeviceInfos)

Field Summary

Fields

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

Constructor Summary

Constructors

Constructor and Description
SMDeviceInfos()

Method Summary

Methods inherited from class java.lang.Object

clone, equals, finalize, 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()
```

com.selligent.sdk

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:

`3.5`

See Also:

`SMManager.sendSMEvent(SMEvent), Serialized Form`

Field Summary

Fields

Modifier and Type	Field and 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 and 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 and Description
boolean	**equals**(java.lang.Object o) Compares this instance with the specified object and indicates if they are equal.
protected com.selligent.sdk.SMEventActionEnum	**getAction**()
protected com.selligent.sdk.SMEvent.Status	**getStatus**()
int	**hashCode**() Returns an integer hash code for this object.
void	**readExternal**(java.io.ObjectInput serializedObject) This method is called when deserializing the object.
protected void	**setStatus**(com.selligent.sdk.SMEvent.Status status)
void	**writeExternal**(java.io.ObjectOutput serializedObject) This method is called when serializing the object.

Methods inherited from class java.lang.Object

clone, finalize, 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 SMEvent object without data and callback

SMEvent

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

Constructs an SMEvent 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

getAction

```
protected com.selligent.sdk.SMEventActionEnum getAction()
```

getStatus

```
protected com.selligent.sdk.SMEvent.Status getStatus()
```

setStatus

```
protected void setStatus(com.selligent.sdk.SMEvent.Status status)
```

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 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 ObjectOutput object used to store the values of the SMEvent object

Throws:

java.io.IOException

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:

`3.5`

See Also:

`SMManager.sendSMEvent(SMEvent), Serialized Form`

Field Summary

Fields

Modifier and Type	Field and Description
`java.lang.String`	**`Email`**

Fields inherited from class com.selligent.sdk.SMEvent

`Callback, Data`

Constructor Summary

Constructors

Constructor and Description
`SMEventUserLogin``()`
`SMEventUserLogin``(java.util.Hashtable<java.lang.String,java.lang.String> data, `**`SMCallback`**` callback)` Constructs a new SMEventUserLogin

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

Constructs a new SMEventUserLogin

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method and Description
boolean	**equals**(java.lang.Object o) Compares this instance with the specified object and indicates if they are equal.
int	**hashCode**() Returns an integer hash code for this object.
void	**readExternal**(java.io.ObjectInput serializedObject) 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 com.selligent.sdk.SMEvent

getAction, getStatus, setStatus

Methods inherited from class java.lang.Object

clone, finalize, 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.lang.ClassNotFoundException
```

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.lang.ClassNotFoundException

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 ObjectOutput object used to store the values of the SMEvent object`

Throws:

`java.io.IOException`

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:

`3.5`

See Also:

`SMManager.sendSMEvent(SMEvent), Serialized Form`

Field Summary

Fields

Modifier and Type	Field and Description
`java.lang.String`	**`Email`**

Fields inherited from class com.selligent.sdk.SMEvent

`Callback, Data`

Constructor Summary

Constructors

Constructor and Description
`SMEventUserLogout``()`
`SMEventUserLogout``(java.util.Hashtable<java.lang.String,java.lang.String> data, `**`SMCallback`**` callback)` Constructs a new SMEventUserLogout

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

Constructs a new SMEventUserLogout

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method and Description
boolean	**equals**(java.lang.Object o) Compares this instance with the specified object and indicates if they are equal.
int	**hashCode**() Returns an integer hash code for this object.
void	**readExternal**(java.io.ObjectInput serializedObject) 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 com.selligent.sdk.SMEvent

getAction, getStatus, setStatus

Methods inherited from class java.lang.Object

clone, finalize, 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.lang.ClassNotFoundException
```

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.lang.ClassNotFoundException

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 ObjectOutput object used to store the values of the SMEvent object`

Throws:

`java.io.IOException`

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:

`3.5`

See Also:

`SMManager.sendSMEvent(SMEvent), Serialized Form`

Field Summary

Fields

Modifier and Type	Field and Description
`java.lang.String`	**`Email`**

Fields inherited from class com.selligent.sdk.SMEvent

`Callback, Data`

Constructor Summary

Constructors

Constructor and Description
`SMEventUserRegister``()`
`SMEventUserRegister``(java.util.Hashtable<java.lang.String,java.lang.String> data, `**`SMCallback`**` callback)` Constructs a new SMEventUserRegister

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

Constructs a new SMEventUserRegister

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method and Description
boolean	**equals**(java.lang.Object o) Compares this instance with the specified object and indicates if they are equal.
int	**hashCode**() Returns an integer hash code for this object.
void	**readExternal**(java.io.ObjectInput serializedObject) 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 com.selligent.sdk.SMEvent

getAction, getStatus, setStatus

Methods inherited from class java.lang.Object

clone, finalize, 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.lang.ClassNotFoundException
```

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.lang.ClassNotFoundException

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 ObjectOutput object used to store the values of the SMEvent object`

Throws:

`java.io.IOException`

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:

`3.5`

See Also:

`SMManager.sendSMEvent(SMEvent), Serialized Form`

Field Summary

Fields

Modifier and Type	Field and Description
`java.lang.String`	**`Email`**

Fields inherited from class com.selligent.sdk.SMEvent

`Callback, Data`

Constructor Summary

Constructors

Constructor and Description
`SMEventUserUnregister``()`
`SMEventUserUnregister``(java.util.Hashtable<java.lang.String,java.lang.String> data, `**`SMCallback`**` callback)` Constructs a new SMEventUserUnregister

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

Constructs a new SMEventUserUnregister

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method and Description
boolean	**equals**(java.lang.Object o) Compares this instance with the specified object and indicates if they are equal.
int	**hashCode**() Returns an integer hash code for this object.
void	**readExternal**(java.io.ObjectInput serializedObject) 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 com.selligent.sdk.SMEvent

getAction, getStatus, setStatus

Methods inherited from class java.lang.Object

clone, finalize, 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.lang.ClassNotFoundException
```

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.lang.ClassNotFoundException

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 ObjectOutput object used to store the values of the SMEvent object`

Throws:

`java.io.IOException`

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 in the foreground for the push from the Selligent Mobile Platform. 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:

`3.5`

Nested Class Summary

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

Constructor Summary

Constructors

Constructor and Description
`SMForegroundGcmBroadcastReceiver``(android.content.Context context)` Constructor of the class.

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method and Description
`android.content.IntentFilter`	**`getIntentFilter`**`()` It creates the IntentFilter that must be used when registering the receiver.
`void`	**`onReceive`**`(android.content.Context context,`

	android.content.Intent intent) This method is called 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

clone, equals, finalize, 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`

com.selligent.sdk

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:

`3.5`

See Also:

`Serialized Form`

Nested Class Summary

Nested Classes

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

Constructor Summary

Constructors

Constructor and 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 and 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	**hasBeenFirstSeenInCurrentSession**() Tells if the SMInAppContent has already been seen or not in the current session.
boolean	**hasBeenSeen**() Tells if the SMInAppContent has already been seen or not.
int	**hashCode**() Returns an integer hash code for this object.
void	**readExternal**(java.io.ObjectInput serializedObject) 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

`clone, finalize, 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 SMInAppContent 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 ObjectOutput 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

com.selligent.sdk

Enum SMInAppContent.DisplayMode

java.lang.Object
    java.lang.Enum<SMInAppContent.DisplayMode>
        com.selligent.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 and Description
OnlyOnce
UntilReplaced

Method Summary

All Methods | **Static Methods** | **Concrete Methods**

Modifier and Type	Method and 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

`clone, compareTo, equals, finalize, 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
```

com.selligent.sdk

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:

`3.5`

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 and Description
`SMInAppContentHtmlFragment()`

Method Summary

All Methods | **Static Methods** | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method and Description
java.lang.String	**getContentCategory**() This method returns the category given at the creation of the instance of SMInAppContentFragment
int	**getContentCount**() This method returns the number of contents in the fragment
SMContentType	**getContentType**() This method returns the type given at the creation of the instance of SMInAppContentFragment
boolean	**hasContent**() Tells if the fragment has content or not.
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
static **SMInAppContentHtmlFragment**	**newInstance**(java.lang.String category, int count) Method used to create a new instance of SMInAppContentHtmlFragment
android.view.View	**onCreateView**(android.view.LayoutInflater inflater, android.view.ViewGroup container, android.os.Bundle savedInstanceState)
void	**onSaveInstanceState**(android.os.Bundle savedInstanceState)
void	**onViewCreated**(android.view.View view, android.os.Bundle savedInstanceState)
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.
void	**show**(androidx.fragment.app.FragmentManager fragmentManager, java.lang.String tag) Display the fragment as a full screen dialog, adding the fragment to the given FragmentManager.
int	**show**(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.

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, startActivity, startActivityForResult, startActivityForResult, startIntentSenderForResult, startPostponedEnterTransition, toString, unregisterForContextMenu

Methods inherited from class java.lang.Object

clone, finalize, 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(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(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 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.

com.selligent.sdk

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:

`3.5`

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 and Description
`SMInAppContentImageFragment()`

Method Summary

All Methods | **Static Methods** | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method and Description
java.lang.String	**getContentCategory**() This method returns the category given at the creation of the instance of SMInAppContentFragment
int	**getContentCount**() This method returns the number of contents in the fragment
SMContentType	**getContentType**() This method returns the type given at the creation of the instance of SMInAppContentFragment
boolean	**hasContent**() Tells if the fragment has content or not.
static **SMInAppContentImageFragment**	**newInstance**(java.lang.String category) Method used to create a new instance of SMInAppContentImageFragment
android.view.View	**onCreateView**(android.view.LayoutInflater inflater, android.view.ViewGroup container, android.os.Bundle savedInstanceState)
void	**onSaveInstanceState**(android.os.Bundle savedInstanceState)
boolean	**onTouch**(android.view.View view, android.view.MotionEvent motionEvent)
void	**onViewCreated**(android.view.View view, android.os.Bundle savedInstanceState)
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.
void	**show**(androidx.fragment.app.FragmentManager fragmentManager, java.lang.String tag) Display the fragment as a full screen dialog, adding the fragment to the given FragmentManager.
int	**show**(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.

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, startActivity, startActivityForResult, startActivityForResult, startIntentSenderForResult, startPostponedEnterTransition, toString, unregisterForContextMenu

Methods inherited from class java.lang.Object

clone, finalize, 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 SMInAppContentImageFragment or null if one of the parameters is null.

Since:

1.4

onTouch

```
public boolean onTouch(android.view.View view,
                       android.view.MotionEvent motionEvent)
```

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(@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(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(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 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.

com.selligent.sdk

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:

`3.5`

See Also:

`SMManager, SMManager.getInAppContents(String, SMContentType, int)`

Method Summary

All Methods	Instance Methods	Abstract Methods

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

Method Detail

onRetrieve

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

com.selligent.sdk

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:

`3.5`

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 and Description
`SMInAppContentUrlFragment()`

Method Summary

All Methods | Static Methods | Instance Methods | Concrete Methods

Modifier and Type	Method and Description
java.lang.String	**getContentCategory**() This method returns the category given at the creation of the instance of SMInAppContentFragment
int	**getContentCount**() This method returns the number of contents in the fragment
SMContentType	**getContentType**() This method returns the type given at the creation of the instance of SMInAppContentFragment
boolean	**hasContent**() Tells if the fragment has content or not.
static **SMInAppContentUrlFragment**	**newInstance**(java.lang.String category) Method used to create a new instance of SMInAppContentImageFragment
android.view.View	**onCreateView**(android.view.LayoutInflater inflater, android.view.ViewGroup container, android.os.Bundle savedInstanceState)
void	**onSaveInstanceState**(android.os.Bundle savedInstanceState)
void	**onViewCreated**(android.view.View view, android.os.Bundle savedInstanceState)
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.
void	**show**(androidx.fragment.app.FragmentManager fragmentManager, java.lang.String tag) Display the fragment as a full screen dialog, adding the fragment to the given FragmentManager.
int	**show**(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.

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, startActivity, startActivityForResult, startActivityForResult, startIntentSenderForResult, startPostponedEnterTransition, toString, unregisterForContextMenu

Methods inherited from class java.lang.Object

clone, finalize, 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(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(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 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.

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:

`3.5`

See Also:

`Serialized Form`

Field Summary

Fields

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

Constructor Summary

Constructors

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

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

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

clone, finalize, 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 SMInAppMessage 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 ObjectOutput 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 SMMapMarker[] getMarkers()
```

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

Returns:

an array of SMMapMarker

Since:

3.4

getButtons

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

Gets the buttons of the SMInAppMessage

Returns:

an array of SMLink

Since:

3.4

getType

```
public com.selligent.sdk.SMMessageType getType()
```

Gets the type of the SMInAppMessage

Returns:

the SMMessageType

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

com.selligent.sdk

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:

`3.5`

See Also:

`SMManager, SMManager.getInAppContents(String, SMContentType, int)`

Method Summary

All Methods | **Instance Methods** | **Abstract Methods**

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

Method Detail

onRetrieve

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

com.selligent.sdk

Enum SMInAppRefreshType

java.lang.Object
    java.lang.Enum<SMInAppRefreshType>
        com.selligent.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:

`3.5`

Enum Constant Summary

Enum Constants

Enum Constant and Description
Daily
Hourly
Minutely
None

Method Summary

All Methods | **Static Methods** | **Concrete Methods**

Modifier and Type	Method and 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

clone, compareTo, equals, finalize, 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`

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:

`3.5`

See Also:

`Serialized Form`

Field Summary

Fields inherited from class com.selligent.sdk.SMNotificationButton

`action, data, id, label, type, value`

Constructor Summary

Constructors

Constructor and 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

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

Constructor Detail

SMLink

```
public SMLink()
```

com.selligent.sdk

Class SMManager

java.lang.Object
    com.selligent.sdk.SMManager

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

Singleton object used to interact with the Selligent Mobile SDK.

Since:

1.0

Version:

3.5

Field Summary

Fields

Modifier and Type	Field and Description
static java.lang.String	**BROADCAST_DATA_BUTTON** **Deprecated.** LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated. Use `SMObserverManager.observeClickedButton(LifecycleOwner, Observer)` instead.
static java.lang.String	**BROADCAST_DATA_GCM_TOKEN** **Deprecated.** LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated. Use `SMObserverManager.observeToken(LifecycleOwner, Observer)` instead.
static 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 `SMObserverManager.observeInAppContents(LifecycleOwner, Observer)` instead.
static 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 `SMObserverManager.observeInAppMessages(LifecycleOwner, Observer)` instead.
static java.lang.String	**BROADCAST_EVENT_BUTTON_CLICKED** **Deprecated.** LocalBroadcastManager being deprecated in latest versions of the androidx library, our broadcasts have been deprecated. Use `SMObserverManager.observeClickedButton(LifecycleOwner, Observer)` instead.
static 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 `SMObserverManager.observeToken(LifecycleOwner, Observer)` instead.
static 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 `SMObserverManager.observeInAppContents(LifecycleOwner, Observer)` instead.
static 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.
static 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).
static 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 `SMObserverManager.observeDismissedMessage(LifecycleOwner, Observer)` instead.
`static 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 `SMObserverManager.observeDisplayedMessage(LifecycleOwner, Observer)` instead.
`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").
`static boolean`	**`DISPLAY_ERROR_MESSAGE`** **Deprecated.** This value is of no use anymore
`static boolean`	**`IS_LOCATIONTRACKER_ACTIF`** **Deprecated.** This is not used
`static java.lang.Class`	**`MAIN_ACTIVITY`** The main activity of your app.
`static float`	**`MIN_DISTANCE_CHANGE_FOR_UPDATES`** **Deprecated.** This is not used
`static long`	**`MIN_TIME_BW_UPDATES`** **Deprecated.** This is not used
`static java.lang.Class`	**`NOTIFICATION_ACTIVITY`** The activity that will be called when opening a notification.
`static java.lang.String`	**`VERSION_LIB`** The version of the Selligent Mobile SDK library

Method Summary

All Methods | Static Methods | Instance Methods | Concrete Methods | Deprecated Methods

Modifier and Type	Method and Description
`boolean`	**`areInAppMessagesEnabled`**`()` This method tells if the management of In App messages is enabled or not
`boolean`	**`areNotificationEnabled`**`()` This method tells if the reception of notifications is enabled or not
`void`	**`checkAndDisplayMessage`**`(android.content.Intent intent, android.content.Context context)` This method will check if there is a message to be displayed in the given intent and will display it.
`void`	**`deleteInAppMessage`**`(java.lang.String messageId)` This method deletes one In-App Message.
`void`	**`deleteInAppMessages`**`(java.lang.String[] messageIds)` This method deletes several In-App Messages at once.
`void`	**`disableGeolocation`**`()` Disables the geolocation functionality.
`void`	**`disableInAppMessages`**`()` This method disables the management of "in app" messages.
`void`	**`disableNotifications`**`()` This method disables the reception of push notifications from the server.
`void`	**`displayLastReceivedRemotePushNotification`**`(android.app.Activity activity)`

	Display the latest received remote notification This is mostly used in conjunction with RemoteMessageDisplayType set to None
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
void	**displayNotification**(android.content.Context context, android.content.Intent intent) This method is to be used only if you set **SMSettings.DoNotListenToThePush** to true.
void	**enableGeolocation**() Enables the geolocation functionality.
void	**enableInAppMessages**(**SMInAppRefreshType** refreshType) This method enables the management of "in app" messages.
void	**enableNotifications**() This method enables the reception of push notifications from the server.
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**.
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**.
java.lang.String	**getGCMToken**() This methods returns the GCM token stored by the SDK after calling **registerDevice(Context)**.
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.
void	**getInAppContents**(java.lang.String category, **SMContentType** type, int max, **SMInAppContentReturn** callbackEvent) Gets the list of valid **SMInAppContent** for the given type and category.
void	**getInAppMessages**(**SMInAppMessageReturn** callbackEvent) Gets the list of all **SMInAppMessage** currently stored by the SDK, unfiltered.
static **SMManager**	**getInstance**()
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
int	**getNotificationIconColor**() Gets the icon color set by setNotificationIconColor.
android.graphics.Bitmap	**getNotificationLargeIcon**() Gets the icon set by setNotificationLargeIcon.
int	**getNotificationSmallIcon**() This returns the resource id set by setNotificationSmallIcon.
SMObserverManager	**getObserverManager**() This method will return the instance of **SMObserverManager** needed for you to listen to the SDK events.
SMRemoteMessageDisplayType	**getRemoteMessagesDisplayType**() Gets the way remote messages are displayed when the app is in foreground.
boolean	**isGeolocationEnabled**() Tells if the geolocation is enabled or not.
void	**registerDevice**(android.content.Context context) **Deprecated.** Use the json file given by the Firebase Console when activating Cloud Messaging on your

	project. The SDK will then automatically retrieve the token.
void	**reload**(**SMSettings** settings, android.app.Activity activity) **Deprecated.** Use **reload(SMSettings, SMCallback)**
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.
void	**sendDeviceInfos**() **Deprecated.**
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
void	**sendEvent**(**SMEvent** event) **Deprecated.** use **sendSMEvent(com.selligent.sdk.SMEvent)** instead
void	**sendSMEvent**(**SMEvent** event) Use this method to send an event to the Selligent platform.
void	**setApplication**(android.app.Application app) This gives to the SDK a pointer to the Application instance.
void	**setFirebaseToken**(java.lang.String token) This method is to be used only if you set **SMSettings.DoNotFetchTheToken** to true.
void	**setInAppContentAsSeen**(**SMInAppContent** inAppContent) Set the given **SMInAppContent** as seen.
void	**setInAppMessageAsSeen**(**SMInAppMessage** inAppMessage) Set the given **SMInAppMessage** as seen.
void	**setInAppMessageAsUnseen**(**SMInAppMessage** inAppMessage) Set the given **SMInAppMessage** as unseen.
void	**setNotificationCallback**(**SMNotificationCallback** callback) **Deprecated.** Since 1.3, a broadcast is performed at the click of a button (cf. **BROADCAST_EVENT_BUTTON_CLICKED**)
void	**setNotificationIconColor**(int argb) This allows the SDK to set a specific color to the icon for the notifications.
void	**setNotificationLargeIcon**(int iconResource) This allows the SDK to use a specific icon for the notifications.
void	**setNotificationSmallIcon**(int iconResource) This allows the SDK to use a specific icon for the notifications.
void	**start**(**SMSettings** settings) Mandatory method used to setup the Selligent Mobile SDK.
void	**start**(**SMSettings** settings, android.app.Application application) Mandatory method used to setup the Selligent Mobile SDK.

Methods inherited from class java.lang.Object

clone, equals, finalize, 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 `SMObserverManager.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 `SMObserverManager.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 `SMObserverManager.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 SMSettings). 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 SMObserverManager.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 SMObserverManager.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 SMObserverManager.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 SMObserverManager.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 `SMObserverManager.observeClickedButton(LifecycleOwner, Observer)` instead.*

String representing a key to retrieve an object inside an intent Use this key to retrive 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 `SMObserverManager.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 NOTIFICATION_ACTIVITY
```

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

MAIN_ACTIVITY

```
public static java.lang.Class 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 SMObserverManager 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 an class extending SMApplication). Any later call with different values would be ineffective. Use this version ONLY if you extend `SMApplication`

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 an class extending SMApplication). Any later call with different values would be ineffective. You must use this version if you do not extend SMApplication

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 -

activity -

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 SMApplication.

Parameters:

app - the Application instance

See Also:

SMApplication

registerDevice

```
@Deprecated
public void registerDevice(android.content.Context context)
```

Deprecated. *Use the json file given by the Firebase Console when activating Cloud Messaging on your project. The SDK will then automatically retrieve the token.*

This method will check if the application just started and, if so, will send a first event to the Selligent Mobile Platform. It will also register the device to Google Cloud Messaging if it isn't already. If the application comes from the background, it doesn't do anything. You have to use this method on the onCreate event of your activities if you do not extend `SMBaseActivity` New since 1.5: if you are using the new Firebase way to retrieve the token (using the json file), then you don't need to call this method. It will not do anything anyway if the GoogleApplicationId is empty.

Parameters:

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

Since:

`1.1`

getGCMToken

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

This methods returns the GCM token stored by the SDK after calling `registerDevice(Context)`. As the registration is asynchronous, the token returned may be empty or not up-to-date if the registration process is not completed.

Returns:

`the stored GCM token`

Since:

`1.3`

See Also:

`registerDevice(Context)`

setFirebaseToken

```
public void setFirebaseToken(java.lang.String token)
```

This method is to be used only if you set `SMSettings.DoNotFetchTheToken` to true. It will store the token in the SDK cache and send it to the Selligent platform if needed (if the SDK doesn't have it already or if the token changed)

Parameters:

`token - The string that you got from Firebase to enable push notifications.`

Since:

`2.1.0`

checkAndDisplayMessage

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

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 `SMBaseActivity`.

Parameters:

`intent - the intent containing the message`

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

Since:

`1.1`

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
```

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`

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:

`1.4`

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 `sendSMEvent(com.selligent.sdk.SMEvent)` instead*
Use this method to send an event to the Selligent platform.

Parameters:

`event - an SMEvent object`

See Also:

`SMEvent`, `SMEventUserLogin`, `SMEventUserLogout`, `SMEventUserRegister`, `SMEventUserUnregister`

sendSMEvent

```
public void sendSMEvent(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:

SMEvent, SMEventUserLogin, SMEventUserLogout, SMEventUserRegister, SMEventUserUnregister

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.

com.selligent.sdk

Class SMMapMarker

java.lang.Object
    com.selligent.sdk.SMMapMarker

All Implemented Interfaces:

`java.io.Externalizable, java.io.Serializable`

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

This class represents a point on a map.

Since:

`3.4`

Version:

`3.5`

See Also:

`Serialized Form`

Constructor Summary

Constructors

Constructor and Description
SMMapMarker()

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method and 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

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

Constructor Detail

SMMapMarker

```
public SMMapMarker()
```

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

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:

`3.5`

See Also:

`Serialized Form`

Field Summary

Fields

Modifier and Type	Field and 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 and Description

SMNotificationButton()

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method and Description
int	**getAction**() The action that will be executed when clicking on the button.
java.lang.String	**getId**()
java.lang.String	**getLabel**()
java.lang.String	**getValue**()
void	**readExternal**(java.io.ObjectInput serializedObject) 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

clone, equals, finalize, 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 ObjectOutput 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

com.selligent.sdk

Interface SMNotificationCallback

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

```
@Deprecated
public interface SMNotificationCallback
```

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:

`3.5`

Method Summary

All Methods | Instance Methods | Abstract Methods | Deprecated Methods

Modifier and Type	Method and Description
`void`	**`onButtonClick`**`(java.lang.String buttonId, int buttonAction, java.lang.String buttonValue)` **Deprecated.**

Method Detail

onButtonClick

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

Deprecated.

com.selligent.sdk

Class SMObserverManager

java.lang.Object
    com.selligent.sdk.SMObserverManager

```
public class SMObserverManager
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:

`3.5`

Constructor Summary

Constructors

Constructor and Description
SMObserverManager()

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method and Description
void	**observeClickedButton**(androidx.lifecycle.LifecycleOwner lifecycleOwner, 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 teh main action of a push is executed.
void	**observeClickedButton**(androidx.lifecycle.LifecycleOwner lifecycleOwner, 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.
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

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

Constructor Detail

SMObserverManager

```
public SMObserverManager()
```

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`

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 boadcast 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 boadcast 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 teh 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

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:

`3.5`

Enum Constant Summary

Enum Constants

Enum Constant and Description
`Automatic`
`None`
`Notification`

Method Summary

All Methods | **Static Methods** | **Concrete Methods**

Modifier and Type	Method and 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

`clone, compareTo, equals, finalize, 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`

com.selligent.sdk

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:

3.5

See Also:

SMManager.start(SMSettings)

Field Summary

Fields

Modifier and Type	Field and Description
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.** You don't need to set this property as long as you use the JSON file given

	by Firebase
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`	**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.
SMThemeCategories	**Theme** **Deprecated.** since 1.4, this value is not used anymore
`java.lang.String`	**WebServiceUrl** The URL of the Selligent web services

Constructor Summary

Constructors

Constructor and Description
SMSettings()

Method Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, 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, DeviceDefaut 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 use the "In app messages" service, we recommend keeping this value to default. * On the other hand, if the application abuse `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 is 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 is 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 is 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()`, `SMManager.displayLastReceivedRemotePushNotification(android.app.Activity)`) Default value is Automatic.

Since:

`1.3`

See Also:

`SMManager`

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 `SMManager.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 `SMManager.displayNotification(android.content.Context, android.content.Intent)` every time a push is received. Use this if you already have your own way to listent 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 `SMManager.enableNotifications()` to allow notifications from Selligent to be sent to the device. Default value is true.

Since:

`3.4.0`

Constructor Detail

SMSettings

```
public SMSettings()
```

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.

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

Since:

1.2

Version:

3.5

Enum Constant Summary

Enum Constants

Enum Constant and 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 and Description
static SMThemeCategories	valueOf(java.lang.String name) **Deprecated.** Returns the enum constant of this type with the specified name.
static SMThemeCategories[]	values() **Deprecated.** Returns an array containing the constants of this enum type, in the order they are declared.

Methods inherited from class java.lang.Enum

clone, compareTo, equals, finalize, 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

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.selligent.sdk.SMMapMarker 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.selligent.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
```

Constant Field Values

Contents

com.selligent.*

com.selligent.sdk.SMManager

Modifier and Type	Constant Field	Value
public static final java.lang.String	BROADCAST_DATA_BUTTON	"SMDataButton"
public static final java.lang.String	BROADCAST_DATA_GCM_TOKEN	"SMDataGCMToken"
public static final java.lang.String	BROADCAST_DATA_IN_APP_CONTENTS	"SMDataInAppContents"
public static final java.lang.String	BROADCAST_DATA_IN_APP_MESSAGES	"SMDataInAppMessages"
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	"3.5.0"