

RELEASE NOTES: IN-APP MESSAGING SDK VERSION 2.3 - Android

These are the main feature releases available in the **In-App Messaging SDK version 2.3 for Android**.

Version 2.3 roll-out: May 21

New functionalities

[Presence enablement for photo sharing - beta](#)

[In-app unread messages badge enablement](#)

[Secure form for in-app messaging](#)

[List of certified and supported devices extended](#)

New APIs

[<!-- handlePushMessage-->](#)

[<!-- getNumUnreadMessages -->](#)

[<!-- Photo sharing-->](#)

Deprecated API

New properties

[<!-- agent PCI bubble-->](#)

[<!-- consumer read status-->](#)

New strings keys

[<!-- Photo Sharing Upload Status -->](#)

[<!-- Accessibility -->](#)

[<!-- Connection status -->](#)

[<!-- PCI -->](#)

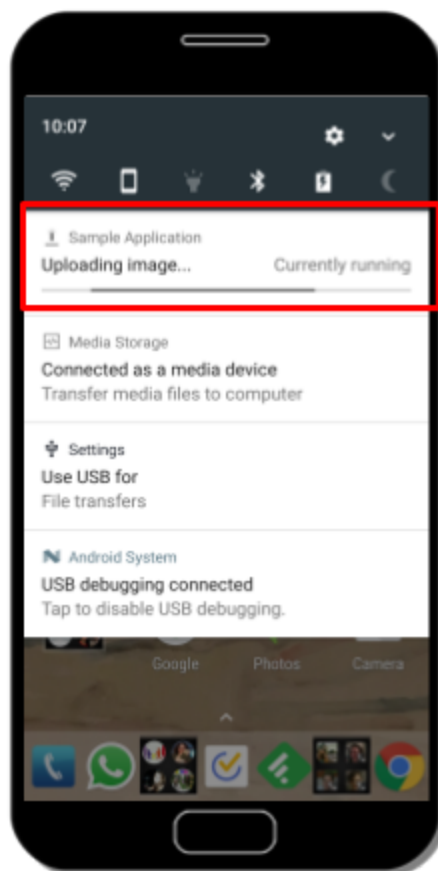
New functionalities

Presence enablement for photo sharing - beta

Presence enablement for photo sharing provides consumers with the ability to receive notifications while uploading a photo, whether they remain within the app or keep it running in the background. The Web Socket remains open for a maximum of 5 seconds when the app moves to the background. This scenario is also applicable for non photo sharing flows.

In addition, brands can also customize notifications for photo sharing, indicating to the consumer when a photo is loading, has successfully sent or has failed to send.

Photo sharing is a beta feature.



Related API: [Photo sharing](#)

Related strings: [Photo sharing upload status](#)

RELEASE NOTES: IN-APP MESSAGING SDK VERSION 2.3 - Android

The following additional conditions and configurations are required*:

Backend update	Backend enablement	Backend configuration	SDK enablement	SDK configuration
N/A	N/A	N/A	N/A	N/A

In-app unread messages badge enablement

When there are unread messages waiting for the consumer within the brand app, this information can be pushed to display in the app's notification badge.

Within the app, brands can develop their own visualisation of a badge, such as a number, icon or other marker to show unread messages.



RELEASE NOTES: IN-APP MESSAGING SDK VERSION 2.3 - Android

Related API: [Handle Push Message](#), [Get Num Unread Messages](#), [Deprecated API](#)

The following additional conditions and configurations are required*:

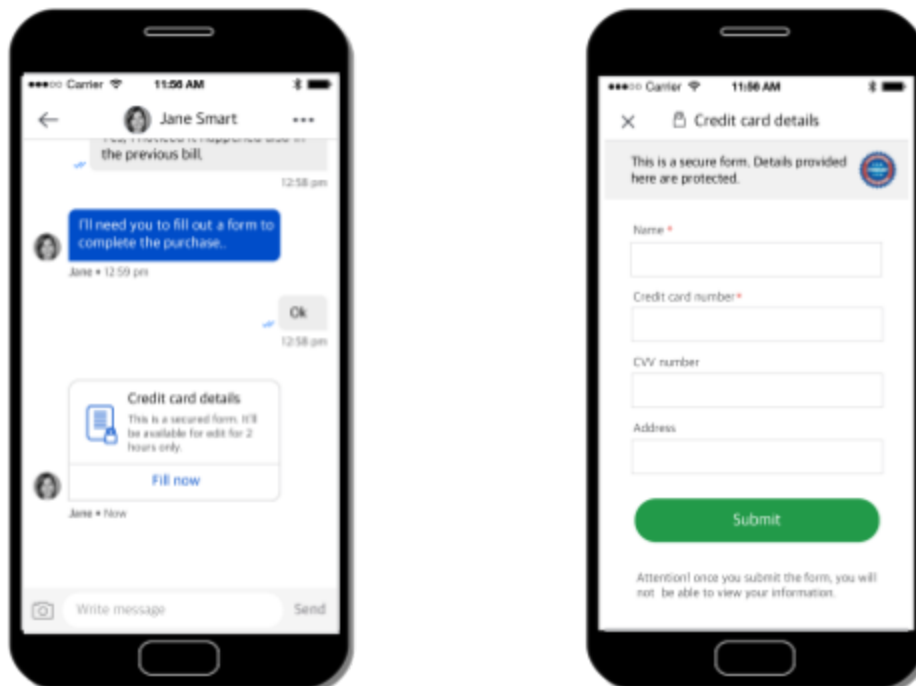
Backend update	Backend enablement	Backend configuration	SDK enablement	SDK configuration
N/A	N/A	N/A	N/A	N/A

Secure form for in-app messaging

The secure form gives consumers the confidence to submit sensitive information, such as credit card data and social security numbers, while messaging in-app. The form also enables agents to safely carry out secure processes, such as payment, identification and authorisations.

The form can be tailored to match the in-app messaging experience and has a time-out expiry, for added security.

*This feature requires consulting services support. For more information, please refer to the LiveEngage [secure form for messaging documentation](#).



Related properties: [Agent PCI bubble](#)

Related strings: [PCI](#)

RELEASE NOTES: IN-APP MESSAGING SDK VERSION 2.3 - Android

The following additional conditions and configurations are required*:

Backend update	Backend enablement	Backend configuration	SDK enablement	SDK configuration
Yes	Yes	Yes	N/A	Yes

List of certified and supported devices extended

The following devices are now also supported and/or certified to host our in-app messaging SDK:

Mobile

	Operating system		
Device	v5.X (Lollipop)	v6.X (Marshmallow)	v7.X (Nougat)
Samsung Galaxy S4	Supported	-	-
Samsung Galaxy A5	Supported	Certified	-
Samsung Galaxy J3	Certified	-	-
Samsung Galaxy J5	-	Certified	-
Samsung Galaxy S6	-	-	Certified
Samsung Galaxy S6 edge	-	-	Certified
Samsung Galaxy S7	-	-	Certified
Samsung Galaxy S7 edge	-	-	Certified
LG V20	-	-	Supported
Sony Xperia X	-	Certified	Supported

RELEASE NOTES: IN-APP MESSAGING SDK VERSION 2.3 - Android

Tablet

	Operating system			
Device	V4.4X (KitKat)	v5.X (Lollipop)	v6.X (Marshmallow)	v7.X (Nougat)
Galaxy Tab A	-	Certified	Supported	-
Galaxy Tab 3	Certified	-	-	-
Nexus	Supported	Certified	Supported	-
HTC	-	-	Supported	Supported

The following additional conditions and configurations are required*:

Backend update	Backend enablement	Backend configuration	SDK enablement	SDK configuration
N/A	N/A	N/A	N/A	N/A

* Key for items as follows:

Backend update: This feature requires an update to the backend.

Backend enablement: This feature requires items to be toggled on in the backend.

Backend configuration: This feature requires configuration in the backend.

SDK enablement: This feature requires items to be toggled on in the SDK.

SDK configuration: This features requires items to be configured in the SDK.

New APIs

<!-- handlePushMessage-->

This API contains all the data received in the push message - Returns PushMessage object.

<i>public static PushMessage handlePushMessage(Context context, Map<String, String> remoteMessage, String brandId, boolean showNotification)</i>	
context	A context from the host app.
remoteMessage	A Map that contains the push message. Push service sends RemoteMessage object - To get the map from this object - call remoteMessage.getData().
brandId	The account ID.
showNotification	Used to instruct the SDK to either show or not show a notification to the user. If you wish your app will handle the display of the notification you can set this as false.

All incoming push messages are received by the host app. The host app can choose to fully handle any push message and display a notification message, or partially handle it and allow the SDK to display the notification.

In a case the host app decides to show its own custom notification, it can call handlePushMessage() with the showNotification parameter set to false and will parse and return a PushMessage object.

In case the push message is not related to the SDK, it will return null.

Note: For the unread messages feature to work correctly, the host app must call this method upon receiving SDK push messages (whether showing a custom notification or not).

<!-- getNumUnreadMessages →

<i>public static int getNumUnreadMessages(String brandId)</i>	
brandId	The account ID.

This returns the counter of the unread messages to the number of push messages received. This number is set to 0 when opening the conversation screen.

RELEASE NOTES: IN-APP MESSAGING SDK VERSION 2.3 - Android

To get updates on the unread messages counter: create a BroadcastReceiver that will listen to the following Action:

LivePerson.ACTION_LP_UPDATE_NUM_UNREAD_MESSAGES_ACTION;

To get the number of unread messages out of the intent, use the following extra key:

LivePerson.ACTION_LP_UPDATE_NUM_UNREAD_MESSAGES_EXTRA;

<!-- Photo sharing-->

Set a pending intent for the image upload foreground service notification -

setImageServicePendingIntent(PendingIntent pendingIntent)

Set a notification builder for the image upload foreground service notification -

setImageServiceUploadNotificationBuilder(Notification.Builder builder)

Set a notification builder for the image down foreground service notification -

setImageServiceDownloadNotificationBuilder(Notification.Builder builder)

Deprecated API

The following API has been deprecated:

```
public static void handlePush(Context context, Bundle data, String brandId, boolean showNotification)
```

Please use the above [handlePushMessage\(\)](#) method instead.

New properties

The following properties of the secure form bubble on the agent side can now be configured:

<!-- agent PCI bubble-->

Background color of the form invitation bubble -

```
<color name="agent_bubble_pci_form_invitation_background_color">  
@android:color/white</color>
```

Background color of the form invitation button only -

```
<color name="agent_bubble_pci_form_invitation_background_btn_color">  
@android:color/white</color>
```

Color of the stroke (border) of the form invitation bubble -

```
<color name="agent_bubble_pci_form_invitation_stroke_color"> @color/lp_gray</color>
```

Color of the text on the button -

```
<color name="agent_bubble_pci_form_invitation_button_text_color">  
@color/lp_blue</color> color of the text on the button
```

Text color on the description in the form invitation bubble -

```
<color name="agent_bubble_pci_form_invitation_description_text_color">  
@color/lp_gray</color>
```

Text color on the title in the form invitation bubble -

```
<color name="agent_bubble_pci_form_invitation_title_text_color">  
@android:color/black</color>
```

Color of the icon in the form invitation bubble -

```
<color name="agent_bubble_pci_form_invitation_icon_tint_color">  
@color/lp_blue</color>
```

The following properties of the status notifications on the consumer side can now be configured:

<!-- consumer read status-->

Color of the read status indicator in the consumer message status line -

```
<color name="consumer_bubble_read_status_color">@color/lp_blue</color>
```

Color of the received status indicator in the consumer message status line -

```
<color name="consumer_bubble_received_status_color">#cecece</color>
```

Color of the sent status indicator in the consumer message status line -

```
<color name="consumer_bubble_sent_status_color">#cecece</color>
```

Color of the sending status indicator in the consumer message status line -

```
<color name="consumer_bubble_sending_status_color">@color/lp_gray</color>
```

New strings keys

<!-- Photo Sharing Upload Status -->

```
<string name="uploading_image">Uploading image...</string>
<string name="downloading_image">Downloading image...</string>
```

<!-- Accessibility -->

```
<string name="lp_accessibility_message_preview_close_description">Close</string>
<string name="lp_accessibility_scroll_down_indicator_description">scroll
down</string>
```

<!-- Connection status -->

```
<string name="lp_connection_status_trying_to_connect">Still trying to
connect...</string>
<string name="lp_connection_status_no_connection">Offline. Please check your
connection.</string>
```

<!-- PCI -->

```
<string name="lpmessaging_ui_pci_leave_dialog_title">Are you sure?</string>
<string name="lpmessaging_ui_pci_leave_dialog_description">Once you leave the secure
form, you will not be able to access it again.</string>
<string name="lpmessaging_ui_pci_leave_dialog_btn_positive">OK</string>
<string name="lpmessaging_ui_pci_leave_dialog_btn_negative">STAY</string>
<string name="lpmessaging_ui_fill_in_form_text_button">Fill in form</string>
<string name="lpmessaging_ui_secure_form_to_fill_in_message">This is a secure form.
Information entered here is protected and cannot be accessed once submitted.</string>
<string name="lpmessaging_ui_secure_form_viewed_message">This form has already been
viewed. Please ask the agent to resend the form.</string>
<string name="lpmessaging_ui_secure_form_error_message">There was a problem opening
this form. Please ask the agent to resend the form.</string>
<string name="lpmessaging_ui_secure_form_expired_message">Secure form has expired.
Please ask the agent to resend the form.</string>
<string name="lpmessaging_ui_secure_form_submitted_message">This form has been
submitted and cannot be reopened for security reasons.</string>
```

RELEASE NOTES: IN-APP MESSAGING SDK VERSION 2.3 - Android

```
<string name="lpmessaging_ui_secure_form_consumer_submitted_message">I have submitted  
the %s form</string>
```

This document, materials or presentation, whether offered online or presented in hard copy ("LivePerson Informational Tools") is for informational purposes only. LIVEPERSON, INC. PROVIDES THESE LIVEPERSON INFORMATIONAL TOOLS "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

The LivePerson Informational Tools contain LivePerson proprietary and confidential materials. No part of the LivePerson Informational Tools may be modified, altered, reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), without the prior written permission of LivePerson, Inc., except as otherwise permitted by law. Prior to publication, reasonable effort was made to validate this information. The LivePerson Information Tools may include technical inaccuracies or typographical errors. Actual savings or results achieved may be different from those outlined in the LivePerson Informational Tools. The recipient shall not alter or remove any part of this statement.

Trademarks or service marks of LivePerson may not be used in any manner without LivePerson's express written consent. All other company and product names mentioned are used only for identification purposes and may be trademarks or registered trademarks of their respective companies. LivePerson shall not be liable for any direct, indirect, incidental, special, consequential or exemplary damages, including but not limited to, damages for loss of profits, goodwill, use, data or other intangible losses resulting from the use or the inability to use the LivePerson Information Tools, including any information contained herein.

© 2017 LivePerson, Inc. All rights reserved.