App Note: Message Property

06/05/2019

**Audience**

This document gives an outline of the message property in terms of design consideration and how-to-use guidance. It is intended for app developers to understand and troubleshoot apps using message properties.

**Introduction**

The message property provides some unique capabilities that are not available with other Ayla platform property types. For example, the Ayla String property type and datapoint has a limitation of 1024 characters. The Ayla platform does support the File property type which can be used as an alternative. However, file property datapoints are not accessible once the datapoint was fetched. The message property type is a new property type supporting larger datapoints and allows for repeated downloading, similar to other Ayla property types. The new message property type allows datapoints up to 512 KB in size to be created and fetched multiple times.

Note that while message property type datapoints can be large and that was a key design goal, they can also be small, or "normal" size. The ability to include JSON formatted message content provides flexibility that makes this property type appealing for a variety of applications*.*

**Message management**

To facilitate message creation and retrieval, the SDK introduces a new property class AylaMessageProperty which extends from AylaProperty. Properties with a base type of “message” can be safely converted to AylaMessageProperty.

**Create message datapoint**

As with other property types, the SDK uses AylaMessageProperty.createDatapoint(String value, successListener, errorLisntener) to create a new message datapoint. A typical success response includes the datapoint identifier with a 201 success code.

**Message content vs message value**

Because the message property is especially designed for large datapoints, the AylaMessageProperty.value field of the message datapoint does not store the message content created. Instead, it contains the ID of the created message datapoint. The datapoint ID is then used for fetching the message content. Once retrieved, the message content is available in AylaMessageProperty.messageContent.

Therefore, AylaMessageProperty.getValue() won’t return the message content, instead, it returns a reference to the message content, with a form similar to "/json\_out/79530406-8310-11e9-daf1-7d941092010b", composed of the property name and message datapoint ID separated by a forwarding slash “/”. Once the message property content has been fetched (see below), use AylaMessageProperty.getMessageContent()to return the locally cached message content.

**Get and fetch message content**

The asynchronous call AylaMessageProperty.fetchMessageContent(successListener, errorListener)request will fetch message contents from the cloud. The fetched message contents will be available in the success listener callback and will be locally cached in AylaMessageProperty.messageContent. Internally, this API calls AylaProperty.fetchDatapointWithID(datapointId, successListener, errorListener). It can also be used to fetch the message content if a known datapointId is available from an application.

To get the contents of a message, the SDK also provides a synchronous call AylaMessageProperty.getMessageContent() which returns the message content if it has been cached from a prior fetch request, else it returns null.

**Auto fetch message content**

By default, the message contents are stored in Ayla’s no-SQL DB and must be retrieved in a two-step process. The SDK does not fetch the message contents for the managed message properties. This is because message properties can be very large and retrieving them by default may not be necessary and can lead to a poor user experience due to wrongly downloading unused large message contents. However, after an evaluation of the use case for message properties in the application, it may be desirable to automatically fetch the message contents. Therefore, the SDK provides a Settings class override AylaSystemSettings.autoFetchMessageContent, allowing message contents to be fetched automatically if set to true. This allows for AylaMessageProperty.getMessageContent() to return the cached message content which is updated from the latest message datapoint. AylaMessageProperty.getMessageContent() would return null if the message content has not been fetched from the cloud.

*Note that autoFetchMessageContent is supported via Aura OEM Configurations by including this line in the your\_oem\_configs.aura file:*

|  |
| --- |
| {  "appId": "...",  …  *"autoFetchMessageContent": true,*  …  } |

See the AuraOEMConfiguration file for more detailed app notes regarding the OEM configurations.

**Appendix**

Error codes and messages for message creation

|  |  |  |
| --- | --- | --- |
| Error Code | Error Message | Condition |
| 422 | You must enter a valid json | mimetype is 'application/json' and invalid json value is provided |
| 422 | message datapoints should be less than 512kb | size of the datapoint > 512kb |
| 422 | Base type not supported | When base type of the property is not "message" |
| 404 | Device not found | Invalid device dsn |
| 404 | Property not found | Invalid property name |