

# [Sendbird] Interactive cards in Chat View

Date: 2020-05-06

Prepared By: Solutions Engineering

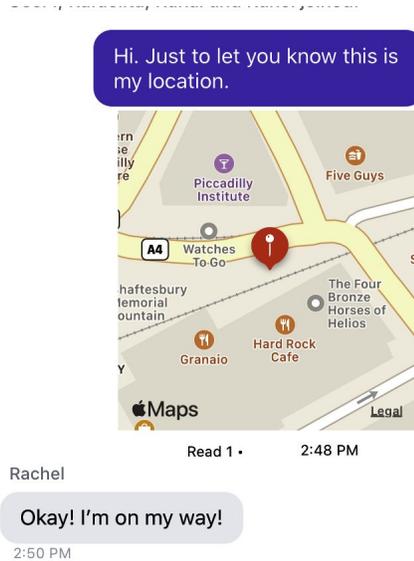
## Background

- This document outlines some considerations for creating interactive message cards in a chat view.
- Interactive message cards can provide useful, fun, and interesting chat interactions for users, they can also help to provide up-to-date dynamic data from external sources.
- There is a short video of interactive quiz message cards in action located [here](#).
- There is an e-commerce card view example code [here](#).

## Considerations

- Dynamic card production
  - Consider producing a fixed number of card types that are occasionally altered or added to an application.
  - For each card type provide display templates that are shipped with your app.
  - Adding or altering the display templates can be more or less dynamic depending on how the code for the display template is delivered to your application.
  - If new card types are frequently needed, consider a mechanism for shipping the new card templates to your application as needed.
  - The display template will sit inside of your message view and render a particular card based on the arrival of a message for that particular card type.
  - It is possible to add a message "custom\_type" (String value) to any sendbird message. Doing so will provide a reference for the application to determine the type of card to render.
    - [iOS](#)
    - [Android](#)
    - [JavaScript](#)
  - See the end of this document for example uses of cards:
- Message creation
  - Consider the information about the card that will be held in the Sendbird message object.
    - Reference only example:

- All of the card generation functionality is stored in the application and the card is triggered to run and display by listening for messages with a particular "custom\_type" or message.
- A location sharing message card is an example of this. The custom type is set to "location" and the message field contains longitude and latitude coordinates only.



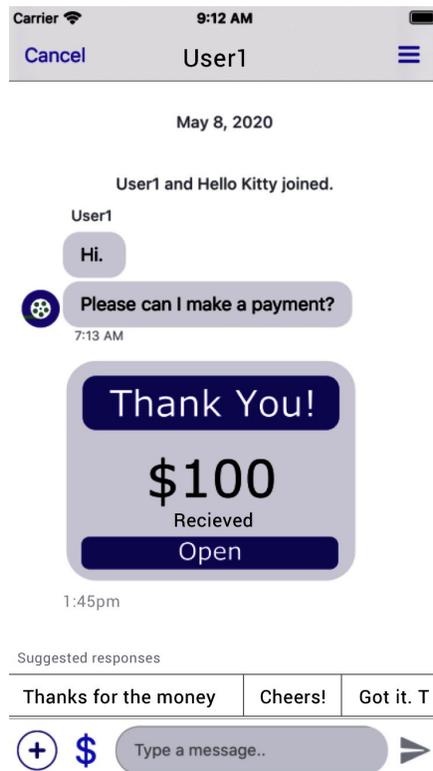
- Another example might be an oversize emoticon. In this example, the message's custom\_type is set to "emoticon" and the data field contains a code for a particular oversize emoticon. The code is used as a reference call for the emoticon gif image.



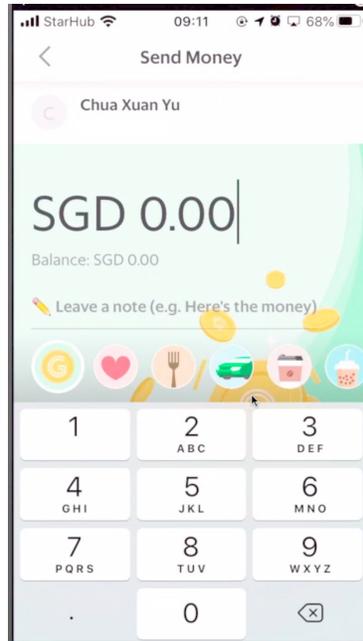
- Everything included in the message option:
  - In this example all of the information to render a card view might be held on the message.
  - This is suitable for simple message cards that contain static data, and the rendering code is small enough to be held in the message object without hindering message download speed or push notifications.
  - The message's data field can contain simple display design data such as styling, fonts, and dimensions, etc, or a rich-text delta.
- Hybrid option:
  - Consider providing a middle way, where some essential static data quickly needed for displaying a message to the user is contained within the Sendbird message object, but then other dynamic data or functionality is stored either in the application or can be fetched from a third source.
- Push notification considerations:
  - When storing data for a card in a Sendbird message consider what if anything about that message will be sent in a push notification.
  - It should be noted that iOS push notifications require a relatively light payload (2kb for iOS and 4kb for FCM and Android). Given there is some required information from sendbird. This leaves a limited amount of room for messages.
    - However, It should be noted that sendbird's push notifications can be sent without including the data field of a sendbird message. This is a global setting and will turn off the data payload for all messages.
- Message field vs Data field vs MetaArrays - [More details](#)
  - Sendbird MESSAGES message objects have a "message" field and a "data" field
  - The "message" field has a limit of 5000 characters or less.
  - The "data" field can hold String type data including serialized JSON objects.
    - The data field's size is not limited by Sendbird. However, consider that the larger the data field the longer it will take to send and receive a message.
  - The "metaArray" field can hold key-value pair data that can be edited by any member of a channel.
- Sendbird card message and third-party APIs.
  - If you are using card messages that will display dynamic data, load as much of the card as possible while fetching data from the third-party API.
  - Consider how to gracefully handle a failure of the 3rd party service, and consider providing an automated service to retry fetching the data.

- Where possible hold the returned data from the 3rd party locally for as long as needed and use that local version ahead of fetching the same data again. This case is useful for when a user is scrolling up and down the message view and the same message may come into view multiple times.

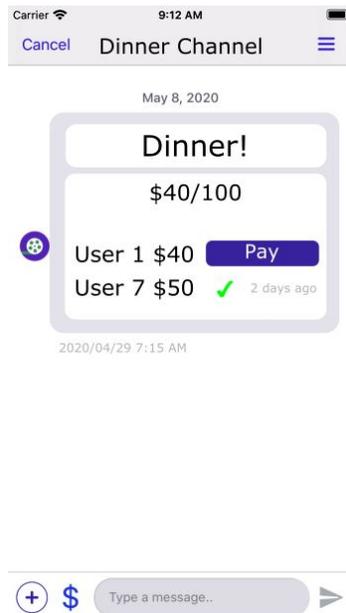
## Peer-to-peer payment receipt



## Peer-to-peer payment action: User can select card background



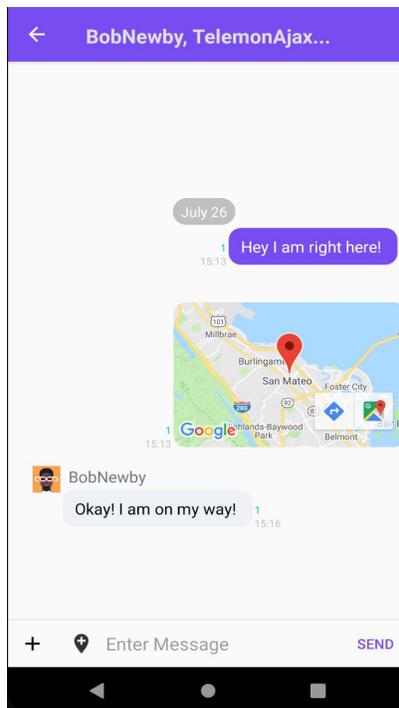
## Peer-to-peer: Split bill



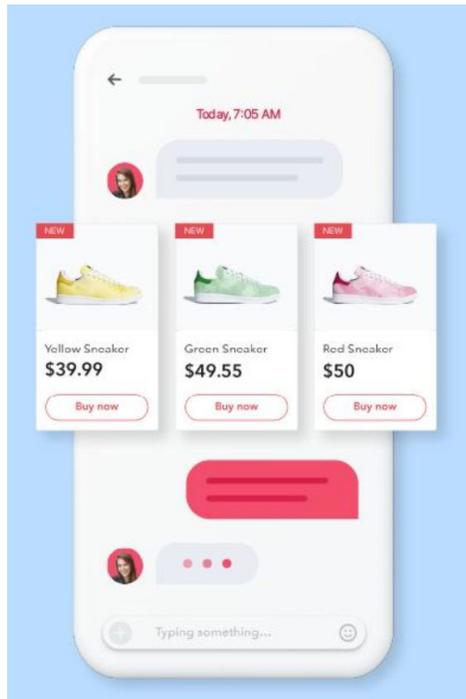
## Oversize Emoji



## Location Sharing



## Commerce



## Sendbird Desk Questionnaire

- Consider the case where a desk agent wants to provide a questionnaire. For example a questionnaire about Covid19 symptoms.
  - First, Agent sends a trigger message which remains hidden always.
    - The trigger message might be a hashtag code such as #sendCovidForm, or the agent could use a pre-defined global auto reply.
  - On the customer's side, the hidden message triggers a question card interface for the current user.
  - The current user completes the question card and sends the completed details as text back to the agent.
    - The agent doesn't see the card. Instead the agent sees the returned questions in plan text form. For example:
      - Fever or chills - YES
      - Cough - YES
      - Shortness of breath or difficulty breathing - NO
      - Fatigue - NO
  - The card interface became a message from the User, and the trigger message always remains hidden.