

General

The general format will be of the pattern `application/vnd.mitel.<type>+<parse-format>` where `<parse-format>` can be either `json` or `xml`. If the `parse-format` isn't supplied, the default is `json`. When an application posts a message with a `contentType` other than `text/plain`, it SHOULD also include the `fallbackText` property as `text/plain`. This allows applications who don't yet understand the `contentType` an alternative to show users.

Recognized types

- **whisper**: indicates that the message is an [ephemeral notification only message](#). It's not available for future gets. It's directed at a single participant within a conversation.
- **attachment**: indicates a link to an attachment in the conversation. Body is the `attachmentId` for the application to fetch the attachment.
- **location**: a post of a location from a client. Body is of the [MDN form](#):

```
{
  "coords": {
    "latitude": "double",
    "longitude": "double"
  }
}
```

- **quoted-message**: allows clients to quote a previous message in the conversation. Body is of the form:

```
{
  "content": {
    "contentType": "text/plain | text/html",
    "body": "string"
  },
  "cite": {
    "messageId": "mess-id-123",
    "createdBy": "string",
    "createdOn"?: "date-time UTC ISO string",
    "body": "string"
  }
}
```

- **action-card**: Extensibility means for third parties to post custom cards into a chat stream to extend functionality. This is a complex type and usage scenarios will be described below. Body is of the form:

```
{
  "contentType": "application/vnd.mitel.action-card",
```

```

"body": {
  "headerImageUrl": "string",
  "title": "string",
  "subTitle": "string",
  "bodyContentType": "text/plain | text/html",
  "body": "string",
  "buttons": [{
    "text": "string",
    "action": "post-message | post-quoted-message | share-location | phone |
http-post | open-link",
    "targetUrl": "string", // link target, phone tel, or http target
    "payload": "string", // request body of a request where applicable
    "behavior": {
      "disableOnClick": "boolean",
      "success": {
        "action": "post-message | post-reaction",
        "payload": "text"
      },
      "failure": {
        "action": "post-message | post-reaction",
        "payload": "text"
      }
    }
  }]
},
"fallbackText" : "text to display when content type isn't understood"
}

```

Action cards

Action cards are a versatile type which allows for a variety of application customizations to suit the use case. Action buttons provide a means to add client behavior to an application. Any application that uses chat can then be extended in a shared manner assuming they follow the spec. The power is driven thru the action buttons. The buttons describe what an application should do when these are invoked. Click behaviors include, but aren't limited to, message posting, generic http posting, opening links, and launching phone calls. Then response behaviors can optionally be identified as to how to response to success of the action. A few examples will demonstrate some sample usages.

Examples

Post bodies are stringified JSON. The examples below will show expanded JSON for readability. Apologies for the ascii art but inline images aren't supported in the github wiki. Trailing comments are notes related to the line.

Quick replies Question asked and two buttons are presented. Both submit payload back to conversation. If no payload is specified then the title is used for the button label

```

{
  "contentType": "application/vnd.mitel.action-card",
  "body": {

```

```

"body": "What's for dinner?",
"bodyContentType": "text/plain",
"buttons": [
  {
    "title": "Tacos",
    "action": "post-quoted-message"
  },
  {
    "title": "Hot dogs",
    "action": "post-quoted-message",
    "payload": "hotdogs"
  }
]
}

```

This would display:

```

User 1:  What's for dinner?
        -----
        | Tacos | | Hot dogs |
        -----
User 2:  Tacos // message posted when the first button is clicked.

```

Advanced action buttons Buttons can be combine together to do a number of things.

```

{
  "contentType": "application/vnd.mitel.action-card",
  "body": {
    "bodyContentType": "text/html",
    "body": "Action buttons example. Click to perform an action.",
    "buttons": [
      {
        "title": "Call me",
        "action": "phone",
        "targetUrl": "tel:+16135551234",
        "behavior": {
          "success": {
            "action": "post-message",
            "payload": "Phone button clicked"
          }
        }
      },
      {
        "title": "Google CloudLink",
        "action": "open-link",
        "targetUrl": "http://lmgty.com/?q=cloudlink",
        "success": {
          "action": "post-message",

```

```

        "payload": "Link clicked"
      }
    }
  },
  {
    "title": "Where are you?",
    "action": "share-location"
  },
  {
    "title": "HTTP post",
    "action": "http-post",
    "targetUrl": "http://httpbin.org/post",
    "payload": {
      "var1": "data",
      "var2": "data"
    },
    "success": {
      "action": "post-message",
      "payload": "HTTP post clicked"
    }
  }
]
}
}

```

This would display:

User 1: Action buttons example. Click to perform an action.

```

-----
| Call me | | Google CloudLink | | Where are you? | | HTTP post |
-----

```

User 2: Phone button clicked // message posted when "Call me" is clicked. Client should attempt to launch a phone call.

User 2: Google CloudLink clicked // message posted when "Google CloudLink" is clicked. Client should open the page.

User 2: (lat, long) // message posted when "Where are you?" is clicked. This is the fallback text. GUI should render the location showing something like google maps.

User 2: HTTP post clicked // message posted when "HTTP post" is clicked. This would invoke the post which could do any number of things.

Action card The full usage of the action card allows for customized simple actions as a measure of extensibility without deep integrations. For example, a customer can use workflow to create an integration to a inhouse system. It could post an action card which would let appropriate users respond without having to leave the chat system. An example could look like this:

```

{
  "contentType": "application/vnd.mitel.action-card",
  "body": {

```

```
    "headerImageUrl": "https://www.mitel.com/-/media/mitel/images/content-entry-photos/photo-header-table-top-view-of-male-coworkers-analyzing-computer-data.jpg?modified=20180411211619",
    "title": "CloudLink collaboration",
    "subTitle": "Available for purchase",
    "bodyContentType": "text/plain",
    "body": "Do you need help?",
    "buttons": [
      {
        "title": "Yes",
        "action": "post-quoted-message"
      },
      {
        "title": "Call us",
        "action": "phone",
        "targetUrl": "555-555-5555"
      }
    ]
  }
}
```

This would look something like: [image](#)

```
bot:    image (above)
        CloudLink collaboration
        Available for purchase
        Do you need help?
        -----
        | Yes | | Call us |
        -----
User 1: Yes
```