Attachment

Description

An attachment is rich content attached to a message. It can include content like dynamic widgets, static HTML, FlockML, images and files. Attachments can also have their own buttons.

To see some examples of attachments see Sending Attachments.

Structure of an Attachment

The code block below shows in complete detail every element that can go inside an attachment. Examples of creating different kinds of attachments is covered under Sending Attachments.

An example attachment

```
{
    "id": "<id>",
    "title": "<attachment title>",
    "description": "<attachment description>",
    "appId": "<app id>",
    // Precedence order of previews:
    // widget > html > flockml > images
    "color": "#0ABE51",
    "views": {
        // Attachment widget
        "widget": { "src": "<widget url>", "width": 400, "height": 400 },
        // Inline html
        "html": { "inline": "<inline html>", "width": 400, "height": 400 },
        "flockml": "<inline flockml>",
        // For image, only "src" is mandatory, everything else is optional
        "image": {
            "original": { "src": "<image 1 url>", "width": 400, "height":
400 },
            "thumbnail": { "src": "<image 2 url>", "width": 100, "height":
100 },
            "filename": "foo-bar.png"
        }
    },
    "url": "<unfurled url>",
    "forward": "true",  // default: false
    // For downloads, only "src" is mandatory, everything else is optional.
    // The client can use mime to show an appropriate file icon to the
user.
    // Slashes and dots from the filename should be stripped.
    "downloads": [
        { "src": "<download 1 url>", "mime": "<mime type>", "filename":
"<filename 1>", "size": <bytes> },
        { "src": "<download 2 url>", "mime": "<mime type>", "filename":
"<filename 2>", "size": <bytes> }
    ],
    "buttons": [ {
        "name": "<button 1>",
```

```
            "icon": "<icon 1 url>",
            "action": { "type": "openWidget", "desktopType": "modal",
    "mobileType": "modal", "url": "<action url>" },
            "id": "<button id 1>"
        }, {
            "name": "<button 2>",
            "icon": "<icon 2 url>",
            "action": { "type": "openWidget", "desktopType": "sidebar",
    "mobileType": "modal", "url": "<action url>" },
            "id": "<button id 2>"
        }, {
            "name": "<button 3>",
            "icon": "<icon 3 url>",
            "action": { "type": "sendEvent" },
            "id": "<button id 3>"
        }]
    }
```

Object Attributes

Name	Type	Required	Description
`id`	String	No	A unique identifier for the attachment as provided by your app.
`appId`	String	No	App id for the app that sent the attachment. Any value that your app provides for this attribute will be overwritten with your app's actual id by Flock.
`title`	String	No	The title of the attachment.
`description`	String	No	A longer description of the attachment.
`color`	String	No	A hex value (e.g. "#0ABE51") for the color bar.
`views`	Views	No*	Provides user visible views for the attachment. See below for more details.
`url`	String	No	The URL to open when user clicks an attachment, if no widget or FlockML is provided. When generating a URL Preview this should always be set.
`forward`	Boolean	No	If _true_, the attachment can be forwarded. Default value is _false_.
`downloads`	Array	No*	An array of download objects. **Note**: As of now this array should contain at max one object. Each object has the following attributes: • `src` (_String_, **Required**) – URL of the file • `mime` (_String_) – Mime type of the file (used to select the file icon) • `filename` (_String_) – Name of the file • `size` (_Number_) – Size of the file in bytes

buttons	Array	No	An array of attachment buttons. Each object has the following attributes: • `name` (*String*) – Name of the button. • `icon` (*String*) – URL of the button icon. • `action` (*ActionConfig*) – The action to perform when a user clicks on the button. • `id` (*String*) – An opaque string provided by your app to uniquely identify the button. The id is sent as an attribute of the client. pressButton event.

* While neither `views` nor `downloads` is required, one of these must always be present in the attachment.

Views

Flock will select one of the objects inside `Views` to display on the chat screen. See Sending Attachments for the precedence order of these views.

Name	Type	Required	Description
widget	Object	No*	Displays an attachment widget inside the chat screen in desktop, or pops up a modal when the attachment is opened on mobile. • `src` (*String*) – URL for the widget • `width` (*Number*) – Width of the widget • `height` (*Number*) – Height of the widget
html	Object	No*	Displays the HTML string inside the chat screen in desktop (using an iframe). It has the following attributes: • `inline` (*String*) – The HTML string • `width` (*Number*) – Width of the content • `height` (*Number*) – Height of the content
flockml	String	No*	A string containing FlockML content. It is displayed inside the chat screen on both desktop and mobile.
image	Object	No*	An image for the attachment. This is an object with two attributes: • `original` (*Image*, **Required**) – the full size image • `thumbnail` (*Image*) – a thumbnail for the image • `filename` (*String*) – original file name for the image, if any The value for both these attributes is an `Image` object, described below.

* At least one of these objects must be present

Image

Name	Type	Required	Description
src	String	Yes	The URL of the image

width	Number	No	Width of the image in pixels
height	Number	No	Height of the image in pixels

ActionConfig

This object describes the action that should be triggered when a user clicks on an attachment button.

It can be used to open a widget:

```
{
    "type": "openWidget",
    "url": "https://example.com/widget",
    "desktopType": "sidebar",
    "mobileType": "modal"
}
```

Or open a URL in the browser:

```
{
    "type": "openBrowser",
    "url": "https://example.com",
    "sendContext": false
}
```

Or send an event to the event listener URL:

```
{
    type: "sendEvent"
}
```

It requires at least one attribute, `type`:

Name	Type	Required	Attribute Description
type	String	Yes	The configured action. It can have any one of the following values: • `openWidget` • `openBrowser` • `sendEvent`

In addition, if `type` equals `openWidget`, these attributes may be required:

Name	Type	Required	Attribute Description
url	String	Yes	The widget URL
desktopType	String	Yes	How the widget should be opened on the desktop. It can be either `modal` or `sidebar`.
mobileType	String	No	How the widget should be opened on mobile. There is only one possible value for this, `modal`, which is also the default.

If `type` equals `openBrowser`, these attributes may be required:

Name	Type	Required	Attribute Description
url	String	Yes	The URL to open in the browser
sendContext	Boolean	No	Whether context should be sent to the browser or not. Defaults to `false`.