AMI Openfin Fdc3 Documentation

Last Modified: 9/21/2023

To start utilizing the integration features, you need to run this in the Openfin workspace and include this in your local.properties file:
ami.guiservice.plugins=com.f1.ami.web.guiplugin.AmiWebGuiServicePlugin_OpenFin

New AMI Object

__OpenFin: You can see this in Dashboard->Session Variables.

Supported AMI Script Methods

raiseIntent(String intent, Map context)

- Raises the specified intent with context, to be handled by another app.
    - String context= "{ type: 'fdc3.instrument', id: { ticker: 'AAPL}' } }";
    - map m = parseJson(context);
    - __OpenFin.raiseIntent("ViewChart", m);

broadcast(Map context)

- Broadcasts context to all the other apps in workspace that are in the same color channel.
- Example usage
    - String context= "{ type: 'fdc3.instrument', id: { ticker: 'AAPL}' } }";
    - map m = parseJson(context);
    - __OpenFin.broadcast(m);

addIntentListener(String intentType)

- This method adds an intent handler for the specified intent. You will need to have a corresponding entry in the intent section of the apps.json as well to indicate AMI can handle the specified intent.
- Example usage
    - __OpenFin.addIntentListener("ViewChart");

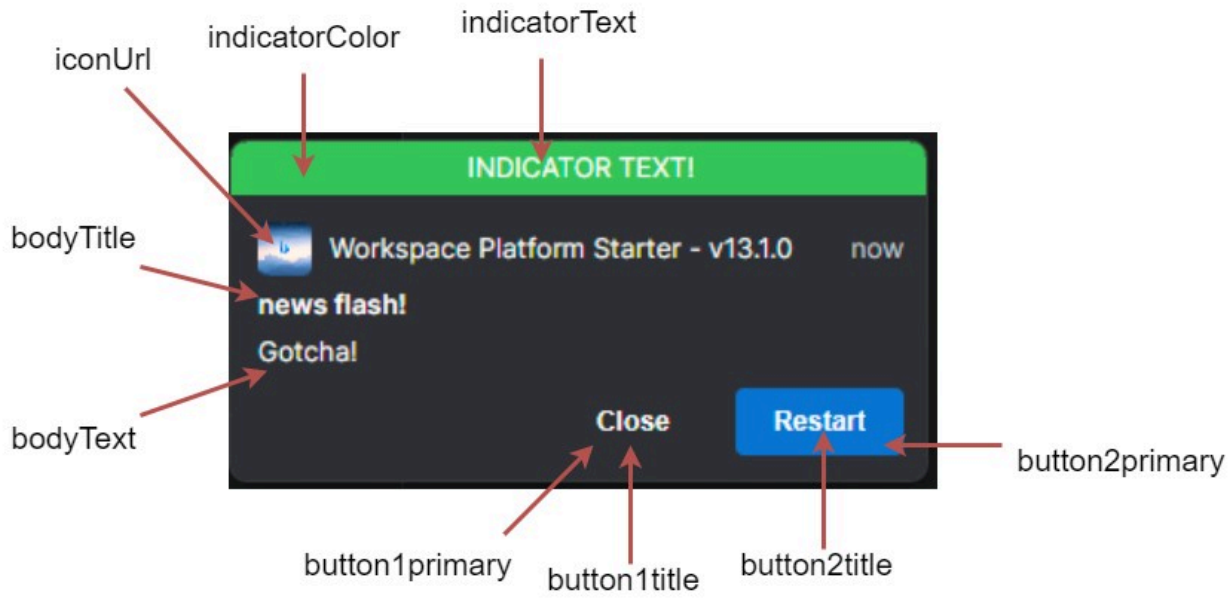
addContextListener(String context type)

- This method qualifies AMI to receive the specified context from a broadcast. Both the broadcaster and the receiver need to be in the same color channel.
- Example usage
    - __OpenFin.addContextListener("fdc3.instrument");

sendNotification(Map options)

- Sends a notification to the Openfin workspace.
- Available keys in the map are:
    - bodyTitle (required)
    - bodyText (required)
    - iconUrl
    - customData
        - You may specify the data for each notification, to be used when user clicks on a button
    - indicatorColor
        - Must be of the following: red, green, yellow, blue, purple, gray
    - indicatorText
    - button1title (required if adding buttons, see below for explanation)
    - button1primary
    - button1data
- For adding button, follow this format:
    - button + number + title/data/primary/iconUrl
        - number corresponds to the ordering of the button. The lower the value, the more left the button appears.
            - E.g. if you have 3 buttons, 1-3, the ordering will be 1 2 3.
        - primary indicates whether it will have the blue background color, this color cannot be changed.
        - title: title of the button, required.
        - iconUrl: image url for the button icon.
        - data: custom data for each button.
- Note that the maximum number of buttons is 8, per OpenFin.
- Example configuration below creates 2 buttons:
    - Map config= new Map();
    - config.put("customData",new map("myData", "data1"));
    - config.put("indicatorColor","green");
    - config.put("indicatorText","indicator Text!");
    - config.put("bodyTitle","news flash!");
    - config.put("bodyText","Gotcha!");
    - config.put("button1title","Close");
    - config.put("button1data","some data here");
    - config.put("button1primary","false");
    - config.put("button2title","Shut Down");
    - config.put("button2data","some data there");
    - config.put("button2title","Restart");

- __OpenFin.sendNotification(config);



*button1iconUrl and button2iconUrl not shown here.

New AMI Script Callbacks

1. **onRaiseIntent(Object intentResolution)**: triggered when AMI receives an intentResolution or its result from raising the intent.
    a. On fdc3 ver 2.0, if AMI is able to get the result of the intentResolution, then you will receive the result, otherwise the intentResolution is returned.
    b. On fdc3 ver 1.2, it always returns the intentResolution.
2. **onContext(Object context, Object metadata)**: triggered when AMI receives a broadcast of a specific context.
    a. You will need to set up a listener first to receive the specific context. See Supported AMI script methods for an example.
3. **onReceiveIntent(Object context)**: triggered when AMI receives an intent from another app.
    a. You will need to set up an intent listener first to trigger this. See Supported AMI script methods for an example.
4. **onNotificationAction(Object event)**: triggered when the user clicks on a button in the notification.
    a. You can use the following to parse the json into a map for ease of access
        i. Map m= parseJson((String) event);
    b. Below is a sample json structure that you will receive from the callback once the user clicks on the button

```
{
    "type": "notification-action",
    "trigger": "control",
    "notification": {
        "form": null,
        "body": "Gotcha!",
        "buttons": [
            {
                "submit": false,
                "onClick": {
                    "data": "some data here"
                },
                "index": 0,
                "iconUrl": "",
                "cta": false,
                "title": "Close",
                "type": "button"
            },
            {
                "submit": false,
                "onClick": {
                    "data": "some data there"
                },
                "index": 1,
                "iconUrl": "",
                "cta": true,
                "title": "Restart",
                "type": "button"
            }
        ],
        "onExpire": null,
        "onClose": null,
```

```
        "onSelect": null,
        "stream": null,
        "expires": null,
        "date": "2023-09-21T18:09:04.481Z",
        "toast": "transient",
        "customData": {
            "myData": "data1"
        },
        "priority": 1,
        "icon":
"http://www.bing.com/sa/simg/facebook_sharing_5.png",
        "indicator": {
            "color": "green",
            "text": "indicator Text!"
        },
        "allowReminder": true,
        "category": "default",
        "title": "news flash!",
        "template": "markdown",
        "id": "a938f456-ef36-4f21-8062-7dd556bf093d"
    },
    "source": {
        "type": "desktop",
        "identity": {
            "uuid": "workspace-platform-starter",
            "name": "f3amione"
        }
    },
    "result": {
        "data": "some data here"
    },
    "control": {
```

```
        "submit": false,
        "onClick": {
            "data": "some data here"
        },
        "index": 0,
        "iconUrl": "",
        "cta": false,
        "title": "Close",
        "type": "button"
    }
}
```