

5.3 Mobile API Plugin for Appery.io Tutorial

Bright Pattern Documentation

Generated: 9/28/2021 6:53 pm

Content is available under license unless otherwise noted.

Table of Contents

Table of Contents	2
Purpose	3
Audience	3
Prerequisites	3
Creating the Project	3
Importing the Plug-in	3
Using the JavaScript API	6
Connecting to an Agent	6
Sending a New Chat Message	7
Receiving Chat Messages	7
Typing Events	7
Uploading an Image	7
Ending the Conversation	7
Showing Forms	8
Testing the App	8

Purpose

The *Mobile API Plug-in for Appery.io Tutorial* will teach you how to use the Mobile API plug-in to connect with the ServicePattern Mobile API.

You will learn how to:

- Link the app with your particular Mobile/Web scenario
- Add a “help me” page to connect with an agent
- Use pre-built user interface (UI) pages for rich contact center interaction

For more information about the API methods and events discussed in this tutorial, as well as other functions supported by the API, see the [Mobile/Web API Specification](#).

[Next >](#)

Audience

This tutorial is intended for the IT personnel responsible for the data infrastructure of the Bright Pattern Contact Center-based contact centers. Readers of this tutorial are expected to have experience in mobile application development as well as a solid understanding of contact center operations.

[< Previous](#) | [Next >](#)

Prerequisites

You must have a [Messaging/Chat scenario entry](#) set up in your Bright Pattern Contact Center configuration. The unique identifier of the scenario entry should match the *appId* value used in the plug-in.

[< Previous](#) | [Next >](#)

Creating the Project

- Click **Create new app**.
- Enter **ServicePattern Mobile App** for the app name.
- Click **Create**.

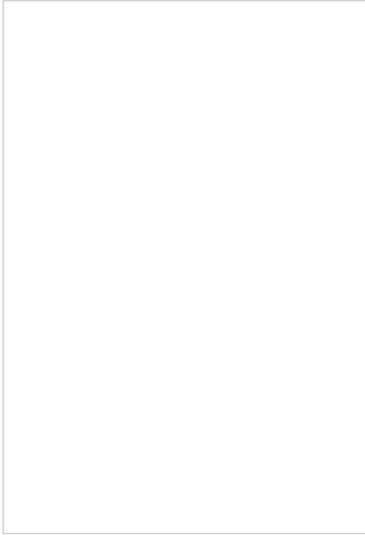
[< Previous](#) | [Next >](#)

Importing the Plug-in

To import the plug-in, follow these steps.

Step 1: Create new from plug-in.

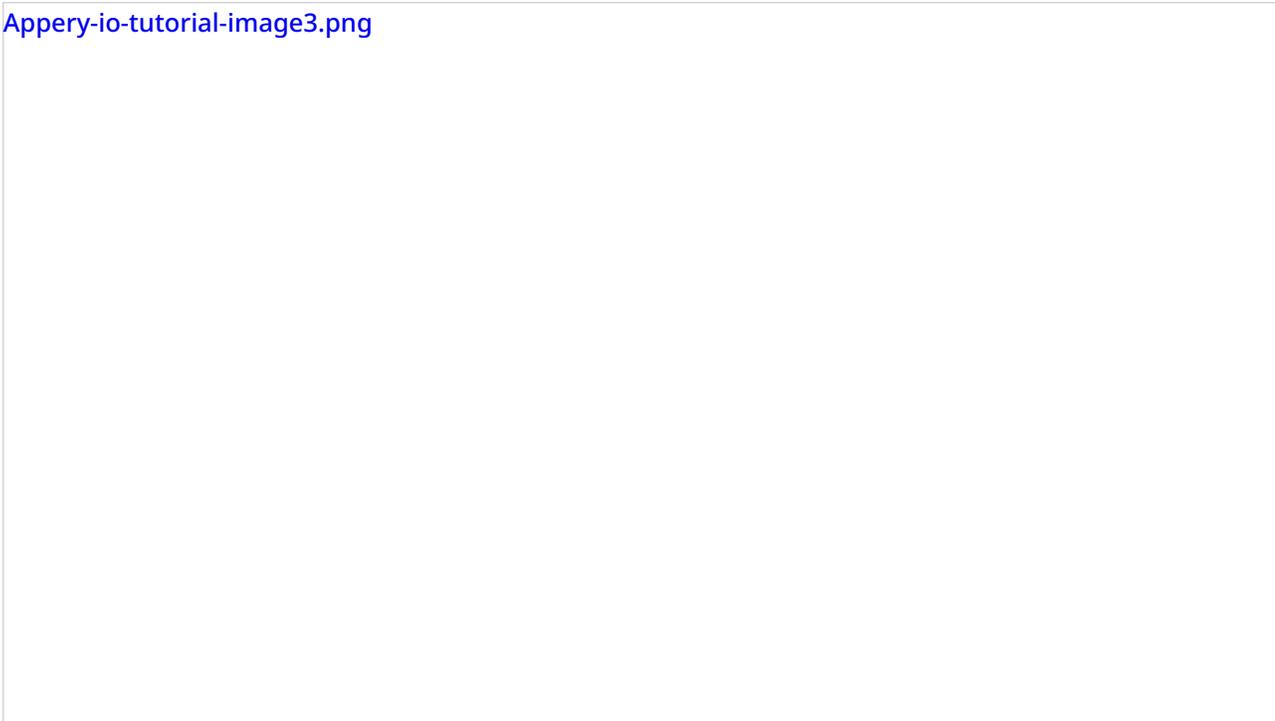
Click **Create New > From Plug-in.**



Step 2: Select the plug-in.

Select the **Bright Pattern > ServicePattern Mobile API** checkbox.

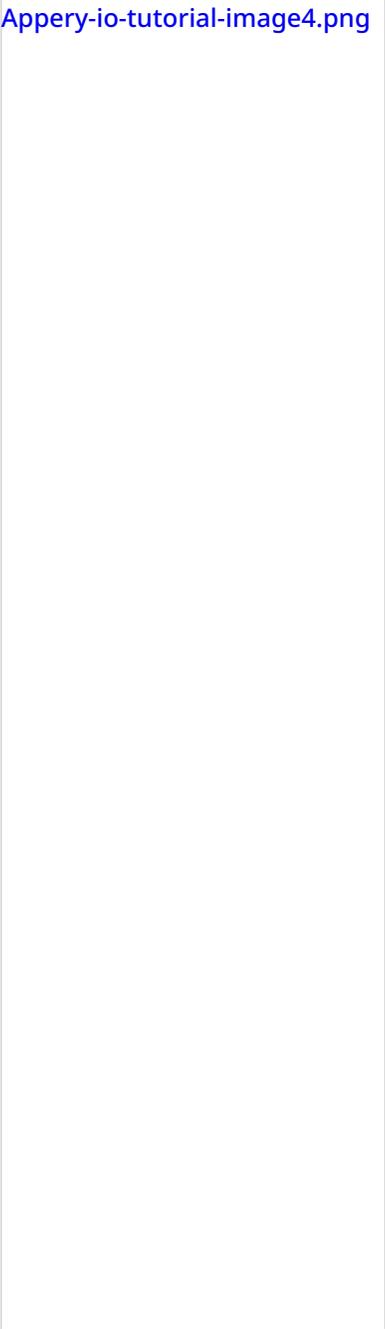
[Appery-io-tutorial-image3.png](#)



Step 3: Import the plug-in.

Click **Import selected plug-ins.** You should see a list of pages and services similar to this one:

Appery-io-tutorial-image4.png



Step 4: Specify values for the plug-in.

- Click **Services > SP_mobileSettings**.
- In the Contact Center Administrator application, copy the *Unique identifier* of your Bright Pattern Contact Center Mobile/Web scenario entry and paste it to *appId*.
- Set *clientId* to a unique value.
- Set the *clientWebServerUrl* value to be the URL of your Bright Pattern Contact Center Client Web Server. You can obtain this URL from your service provider.
- Set the *tenantUrl* value to be your particular tenant URL in Bright Pattern Contact Center . (Tenant URL

corresponds to the domain name of your contact center that you see in the upper right corner of the Contact Center Administrator application after login.)

[Appery-io-tutorial-image5.png](#)



[< Previous](#) | [Next >](#)

Using the JavaScript API

The REST services have already been mapped with the pre-built UI pages provided with the plug-in. However, if you wish to build your own UI or modify the provided UI, you can easily do this by registering your own event listeners with the provided javascript object *SP_mobileAPI*. If you open the *SP_chat* page, you can see the event mappings with the pre-built UI by the following javascript run on load:

```
SP_mobileAPI.addEventListener('chat_session_message', SP_demoUI.onChatMessage );
```

```
SP_mobileAPI.addEventListener('chat_session_party_joined', SP_demoUI.onPartyJoined );
```

```
SP_mobileAPI.addEventListener('chat_session_party_left', SP_demoUI.onPartyLeft );
```

```
SP_mobileAPI.addEventListener('chat_session_typing', SP_demoUI.onAgentTyping);
```

```
SP_mobileAPI.addEventListener('chat_session_not_typing', SP_demoUI.onAgentNotTyping);
```

```
SP_mobileAPI.addEventListener('chat_session_file', SP_demoUI.onImageUploaded);
```

The following sections show you how to set up these mappings and use the javascript API to connect with the ServicePattern Mobile API.

Connecting to an Agent

Once the connection is established, javascript object *SP_mobileAPI.session* will be created. To test if there is an active connection, use the following logic:

```
if( SP_mobileAPI.isConnected() ) {  
  // connected to an agent  
} else {  
  // not connected  
}
```

Sending a New Chat Message

If the connection is active, you can send a new chat message by calling

```
SP_mobileAPI.sendChatMsg(chatMsg);
```

Receiving Chat Messages

First, register an event listener like this one:

```
SP_mobileAPI.addEventListener('chat_session_message', onChatMessage );
```

After that, *onChatMessage* will be invoked whenever a new chat message is received.

Typing Events

When the agent is typing, event *chat_session_typing* occurs. When the agent stops typing, the event *chat_session_not_typing* occurs.

To notify the server when the client is typing, call *SP_mobileAPI.onTyping(true)*.

To notify the server when the client is not typing, call *SP_mobileAPI.onTyping(false)* .

Uploading an Image

If the connection is active, you can call the following function within your app to upload an image:

```
SP_mobileAPI.uploadImage(imageURI);
```

Ending the Conversation

You can end the conversation by calling *SP_mobileAPI.endChat()*.

Showing Forms

To receive the event for showing the form involved with your scenario, register the event listener for *chat_session_form_show*:

```
SP_mobileAPI.addEventListener('chat_session_form_show', onFormShow );
```

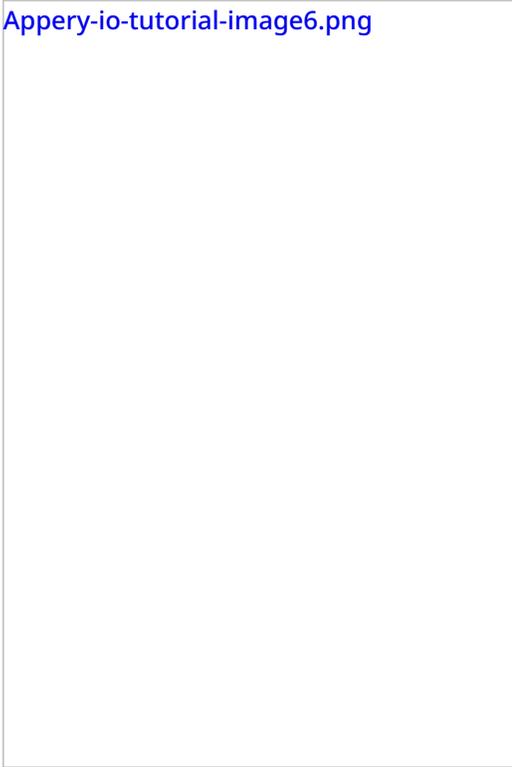
The function *onFormShow* would be called with the parameters (*form_name*, *form_id*) defined by your scenario entry. The demo UI provides a survey form with a form name called *MobileChatSurvey*.

[< Previous](#) | [Next >](#)

Testing the App

Once the app is started, the *SP_Settings* page should appear as follows:

[Appery-io-tutorial-image6.png](#)



The user is prompted to enter a name and phone number. The information would be saved so the user does not have to enter it every time.

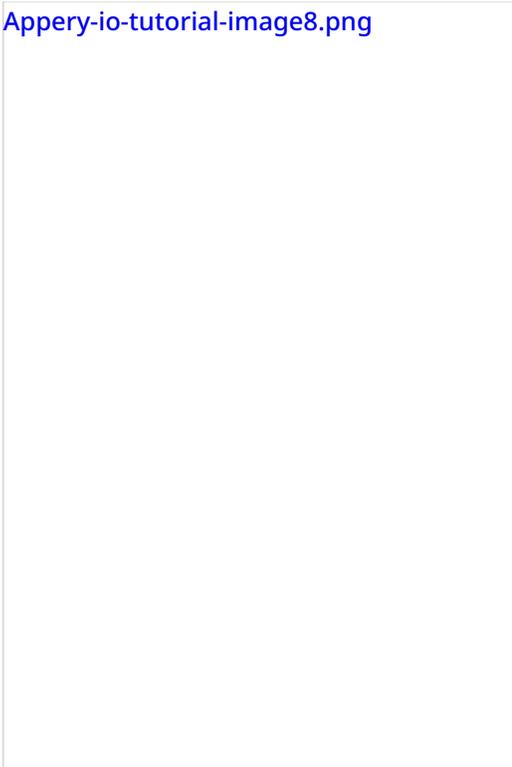
When the **Save** button is clicked, the *SP_home* page should appear:

Appery-io-tutorial-image7.png

When the **Help me** button is clicked, the app should attempt to connect to an available agent through the scenario entry defined by the *appId*. The user has the option to enter the first message and/or upload the image that the user wishes the agent to see.

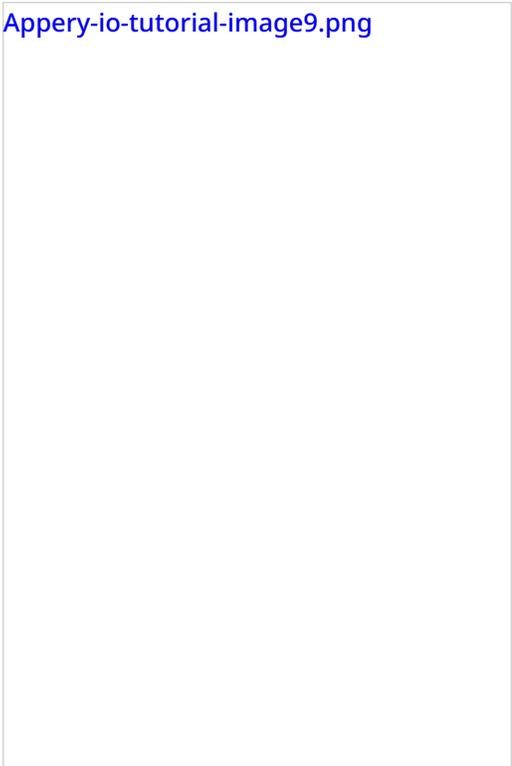
When an agent is successfully found, the app should navigate to the *SP_chat* page, where the interaction between the user and the agent takes place, as shown.

[Appery-io-tutorial-image8.png](#)



The user can chat with the agent and upload an image from the user's phone. After the interaction is finished, a survey form should be displayed (assuming that the scenario requests it), as shown.

[Appery-io-tutorial-image9.png](#)



After the survey form is sent, the whole interaction is considered completed and the app should return to the *SP_home* page.

[< Previous](#)