

Программное управление виджетом

В случае если кнопка управления виджетом по умолчанию не устраивает, она может быть отключена и реализована самостоятельно.

Для этого виджет предоставляет набор методов и событий, используя которые, можно реализовать интерфейс управления виджетом любой сложности.

1. Методы

ОПИСАНИЕ МЕТОДА	КОД
После инициализации виджет находится в скрытом состоянии. Чтобы показать его пользователю, необходимо вызвать метод <code>open()</code> :	<code>widget.open();</code>
Также можно открывать виджет сразу с текстом сообщения:	<code>widget.open('Текст сообщения пользователя');</code> ;
Текст сообщения также можно добавить с помощью метода <code>appendText()</code> :	<code>widget.appendText('Текст сообщения пользователя');</code> ;
Скрыть виджет можно либо кликнув по кнопке закрыть в интерфейсе виджета, либо программно:	<code>widget.close();</code>
Также API предоставляет метод смены состояния <code>toggle</code> , который показывает или скрывает виджет:	<code>widget.toggle();</code>
Если виджет расположен в защищенной зоне и при инициализации использовались <code>credentials</code> , то при выходе из защищенной зоны необходимо в обязательном порядке вызвать метод <code>logout()</code> , чтобы закрыть текущую сессию пользователя и предотвратить неавторизованный доступ к переписке:	<code>widget.logout();</code>
Уничтожение виджета, удаляет чат и кнопку:	<code>widget.destroy();</code>
Управление пуш-токеном для нативных пушей через APNS (iPhone) и FCM (Android).	<code>widget.setIphonePushToken("my-iphone-token");</code> <code>widget.setAndroidPushToken("my-android-token");</code>
Ручное обновление авторизованного клиента из CRM:	<code>widget.refreshClient();</code>

2. События

События на которые можно подписаться используя стандартный для JavaScript интерфейс `EventEmitter`:

```
widget.on('close', function () {  
    console.log(' ');  
});  
  
widget.on('open', function () {  
    console.log(' ');  
});  
  
widget.on('message', function () {  
    console.log(' ');  
});
```

С помощью события `message` например, можно проиграть звук, если виджет скрыт, а пользователю пришло сообщение.

```
widget.on('unread', function (count) {  
    console.log(' ', count);  
});
```

С помощью события `unread` можно показывать пользователю количество непрочитанных сообщений. Событие возникает всякий раз, когда значение меняется. При первоначальном получении списка сообщений, если непрочитанных нет, то события не будет. Предполагается, что подписчик инициализирует счетчик сообщений как 0.

```
widget.on('file-clicked', function (url) {  
    console.log(' ', ':', url);  
}
```

Событие `file-clicked` отправляется только в режиме виджета `mobile` и передает подписанную ссылку на файл, на который нажал пользователь. Ссылка действует ограниченное количество времени. Время жизни настраивается на сервере.