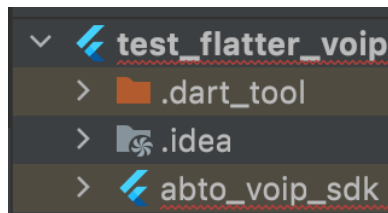


ABTO VoIP Flutter SDK - how to setup and use

1. How to setup Flutter SDK

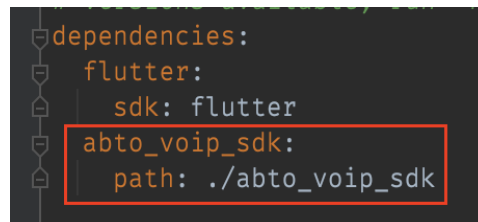
- Put **abto_voip_sdk** folder to your project



```
[YOUR_PROJECT]/abto_voip_sdk
```

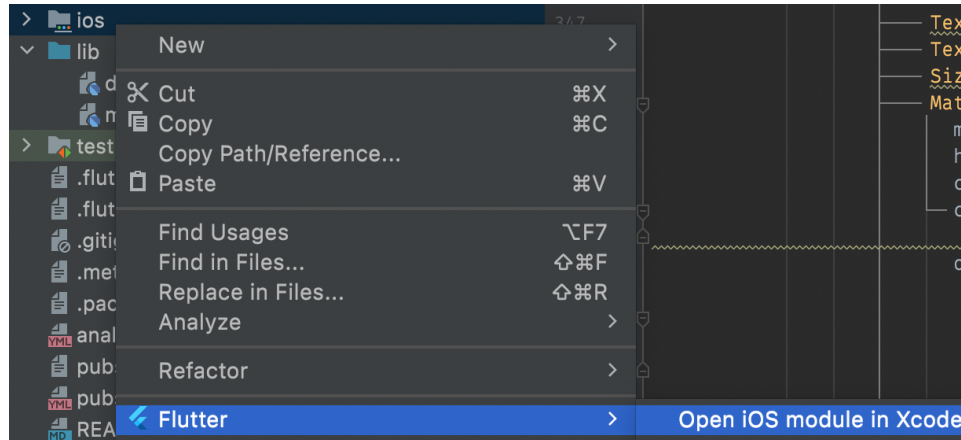
You can find Example Flutter sample inside of plugin:
abto_voip_sdk/example

- Add lib to dependencies in **pubspec.yaml**

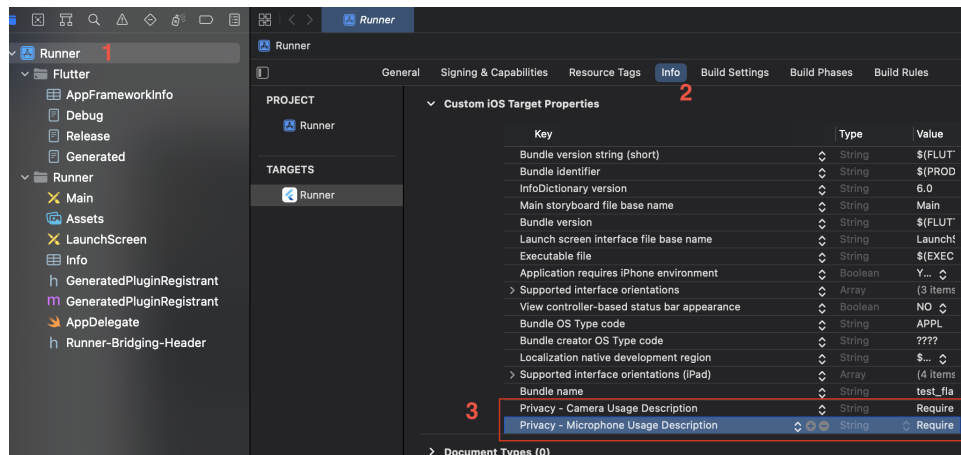


```
abto_voip_sdk:  
  path: ./abto_voip_sdk
```

- Setup IOS part
 - Open iOS part as Xcode project



- Set privacy settings



Privacy - Camera Usage Description: Required for video calls
 Privacy - Microphone Usage Description: Required for audio calls

- Initialise wrapper

```
SipWrapper.wrapper.init();
```

- Set license **userId** and **key** for each platform

```
if (Platform.isAndroid) {  
    SipWrapper.wrapper.setLicense(  
        "{YOUR_ANDROID_USER_ID}",  
        "{YOUR_ANDROID_LICENSE_KEY}"  
    );  
} else if (Platform.isIOS) {  
    SipWrapper.wrapper.setLicense(  
        "{YOUR_IOS_USER_ID}",  
        "{YOUR_IOS_LICENSE_KEY}"  
    );  
}
```

Trial license prefix example:

```
"{Trial...}"
```

```
"{V0exUTjAafwV...}"
```

Commercial license prefix example:

```
"{Licensed...}"
```

```
"{P2FDm7A80...}"
```

You can get trial keys on https://voipsip sdk.com/product/voip-sip-sdk?attribute_pa_platform=multi_mobile

2. How to use Flutter SDK

- Add **sip_wrapper** import to use base methods

```
import 'package:abto_voip_sdk/sip_wrapper.dart';
```

- Initialise wrapper

```
SipWrapper.wrapper.init();
```

- Setup license keys

```
if (Platform.isAndroid) {  
  SipWrapper.wrapper.setLicense(  
    androidLicenseUserId,  
    androidLicenseKey  
  );  
} else if (Platform.isIOS) {  
  SipWrapper.wrapper.setLicense(  
    iosLicenseUserId,  
    iosLicenseKey  
  );  
}
```

- Work with account registering

```
SipWrapper.wrapper.register(login, pass, domain, isSTUNEnabled)  
  
SipWrapper.wrapper.unregister()  
  
SipWrapper.wrapper.registerListener = RegisterListener(  
  onRegistered: () { },  
  onRegistrationFailed: () { },  
  onUnregistered: () { }  
);
```

- Work with call start/end

```
SipWrapper.wrapper.callListener = CallListener(  
  callConnected: (number) { },  
  callDisconnected: () { }  
);  
  
SipWrapper.wrapper.incomingCallListener = IncomingCallListener(  
  onIncomingCall: (number, isVideoCall) { }  
);  
  
SipWrapper.wrapper.startCall(number, isVideoCall)  
  
SipWrapper.wrapper.endCall();  
  
SipWrapper.wrapper.pickUpCall(isVideo);
```

- Options during the call

```
SipWrapper.wrapper.holdStateListener = HoldStateListener(onHoldState: (state) {  
  switch(state) {  
    case HoldState.ACTIVE: break;  
    case HoldState.LOCAL_HOLD: break;  
    case HoldState.REMOTE_HOLD: break;  
  }  
});  
  
SipWrapper.wrapper.hold();  
  
SipWrapper.wrapper.enableSpeaker(true);  
  
SipWrapper.wrapper.mute(true);  
  
SipWrapper.wrapper.startRecord();  
  
SipWrapper.wrapper.stopRecord();  
  
SipWrapper.wrapper.transferCall(number);
```

- Work with text messaging

```
SipWrapper.wrapper.textMessageListener = TextMessageListener(  
  onTextMessageReceived: ( from, to, message) {  
    debugPrint("new message: " + message);  
  },  
  onTextMessageStatus: (address, reason, success) {  
    debugPrint("sent to : " + address + " reason: " + reason);  
  }  
);  
  
SipWrapper.wrapper.sendTextMessage(number, "Hello");
```

- Use video views

```
import 'package:abto_voip_sdk/abto_video_widget.dart';  
  
SizedBox(  
  width: 100,  
  height: 120,  
  child: VoipVideoWidget(true) // true - for outgoing video  
),  
SizedBox(  
  width: 100,  
  height: 120,  
  child: VoipVideoWidget(false) // false - for incoming video  
)
```

- Get SDK version

```
SipWrapper.wrapper.getVersion()
```