Voice Verify - Tutorial: Send a voice message with verification code

🗒️

NOTE:
To add this product to your account, [contact a Telesign expert](). This product is available for full-service accounts only.

This tutorial teaches you how to use Telesign Voice Verify with an SDK, by walking you step-by-step through the creation of an integration where you send your own verification code to an end-user. Skip to the end of this page to see the full, completed [sample code]().

Before you begin

Make sure you have the following before you start:

- **Authentication credentials:** Your Customer ID and API Key. If you need help finding these items, go to the support article [How do I find my Customer ID and API Key]().
- **Testing device:** A mobile phone on which you can receive a call.

Install the SDK

1. Log in to GitHub and choose the [SDK]() in your preferred language.
2. Download or clone the repository. If you download it, extract the repository from the **.ZIP** file.
3. Use these commands to install the SDK:

Node.jsJava (Gradle)Java (Maven)PythonRubyPHPC#

```
npm install telesignsdk -save
```

Send a voice message with a verification code

This section explains how to send a voice message containing a verification code using Voice Verify. The sample code provided earlier is broken out into pieces and explained below.

1. Begin by adding statements for including the appropriate part of the Telesign SDK and any additional language specific functions you may need.

NodeJavaPythonRubyPHPC#

```
var TeleSignSDK = require('telesignenterprisesdk');
```

2. Insert values for each of the items from customer_id/customerId through to phone_number/phoneNumber.

- **customer_id / customerId** - Your Telesign-assigned Customer ID.
- **api_key / apiKey** - Your Telesign-assigned API key.
- **phone_number / phoneNumber** - The phone number to which you want to send the SMS. If you are doing a free trial, the phone number must be verified (verifying the number is not required if you are a paying customer). Include the complete phone number (with country code) including no spaces or special characters.

NodeJavaPythonRubyPHPC#

```
console.log("## verify.voice ##");

const customerId = "FFFFFFFF-EEEE-DDDD-1234-AB1234567890";
const apiKey = "EXAMPLE----
ABC12345yusumoN6BYsBVkh+yRJ5czgsnCehZaOYldPJdmFh6NeX8kunZ2zU1YWaUw/0wV6xfw=="
;
const phoneNumber = "Your test phone number with no special characters or
spaces goes here.";
```

3. Generate a verification code.

NodeJavaPythonRubyPHPC#

```
const optionalParams = {verify_code: "32658"};
```

4. Create a VerifyClient object and pass it your customer_id/customerId and api_key/apiKey.

NodeJavaPythonRubyPHPC#

```
const client = new TeleSignSDK(customerId, apiKey);
```

5. Send a voice message containing the verification code you created. Store the results of your request in

the `response/telesignResponse` variable. For Node.js, use this step to create a callback handler for the voice message you will eventually send and a method called inside voiceCallback that will handle user input.

NodeJavaPythonRubyPHPC#

```javascript
// Callback handler for voice
function voiceCallback(error, responseBody) {
    if (error === null) {
        console.log(`Messaging response for messaging phone number:
${phoneNumber}` +
            ` => code: ${responseBody['status']['code']}` +
            `, description: ${responseBody['status']['description']}`);

        // Ask for user input
        prompt('Enter the verification code received:\n', function (input) {
            if (input === optionalParams['verify_code']) {
                console.log('Your code is correct.');
            } else {
                console.log('Your code is incorrect. input: ' + input + ",
code: " + optionalParams['verify_code']);
            }
            process.exit();
        });

    } else {
        console.error("Unable to send voice call. " + error);
    }
}

// Method to handle user input
function prompt(question, callback) {
    const stdin = process.stdin,
        stdout = process.stdout;

    stdin.resume();
    stdout.write(question);

    stdin.once('data', function (data) {
        callback(data.toString().trim());
    });
}
```

6. Have the end user input the verification code they received, then check it to see if it is valid. Print the results of the check. For Node.js, you send the voice message here, and the callback and user input handler methods you defined in the last step are passed in through the voiceCallback method.

NodeJavaPythonRubyPHPC#

```javascript
// Send Voice request
```

```
client.verify.voice(voiceCallback, phoneNumber, optionalParams);
```

7. If all goes well, you should send a voice message containing a verification code to the phone number you selected. If you are testing with your phone, you should be able to enter the verification code in response to a prompt from your terminal. The sample will then check your verification code to see if it matches the verification code that was sent and report on the results.

Send a Voice Message with Verification Code in a Different Language

This section explains how to send a custom voice message in a different language using Voice Verify. A sample is provided here if you want to cut and paste. Note that the only difference between this code sample and the previous code sample in this tutorial is that you use the tts_message (text-to-speech) parameter to enter a custom message, and the language parameter to specify the language. For a list of language codes, refer to *Supported Languages*.

NodeJavaPythonRubyPHPC#cURL

```
var TeleSignSDK = require('telesignenterprisesdk');

console.log("## verify.voice ##");

const customerId = "FFFFFFFF-EEEE-DDDD-1234-AB1234567890";
const apiKey = "EXAMPLE----
ABC12345yusumoN6BYsBVkh+yRJ5czgsnCehZaOYldPJdmFh6NeX8kunZ2zU1YWaUw/0wV6xfw=="
;
const phoneNumber = "Your test phone number with no special characters or
spaces goes here.";
const optionalParams = {
    tts_message: "Votre code de vérification Widgets 'n' More est $$CODE$$.",
    language: "fr-FR"
};

const client = new TeleSignSDK(customerId, apiKey);

// Callback handler for voice
function voiceCallback(error, responseBody) {
    if (error === null) {
        console.log(`SMS response for phone number: ${phoneNumber}` +
            ` => code: ${responseBody['status']['code']}` +
            `, description: ${responseBody['status']['description']}`);
    } else {
        console.error("Unable to send voice. " + error);
    }
}
```

```
// Send voice request
client.verify.voice(voiceCallback, phoneNumber, optionalParams);
```

Sample code

NodeJavaPythonRubyPHPC#cURL

```javascript
var TeleSignSDK = require('telesignenterprisesdk');

console.log("## verify.voice ##");

const customerId = "FFFFFFFF-EEEE-DDDD-1234-AB1234567890";
const apiKey = "EXAMPLE----
ABC12345yusumoN6BYsBVkh+yRJ5czgsnCehZaOYldPJdmFh6NeX8kunZ2zU1YWaUw/0wV6xfw=="
;
const phoneNumber = "Your test phone number with no special characters or
spaces goes here.";
const optionalParams = {verify_code: "32658"};

const client = new TeleSignSDK(customerId, apiKey);

// Callback handler for voice
function voiceCallback(error, responseBody) {
    if (error === null) {
        console.log(`Messaging response for messaging phone number:
${phoneNumber}` +
            ` => code: ${responseBody['status']['code']}` +
            `, description: ${responseBody['status']['description']}`);

        // Ask for user input
        prompt('Enter the verification code received:\n', function (input) {
            if (input === optionalParams['verify_code']) {
                console.log('Your code is correct.');
            } else {
                console.log('Your code is incorrect. input: ' + input + ",
code: " + optionalParams['verify_code']);
            }
            process.exit();
        });

    } else {
        console.error("Unable to send voice call. " + error);
    }
}

// Method to handle user input
function prompt(question, callback) {
    const stdin = process.stdin,
        stdout = process.stdout;

    stdin.resume();
    stdout.write(question);

    stdin.once('data', function (data) {
        callback(data.toString().trim());
```

```
    });
}

// Send Voice request
client.verify.voice(voiceCallback, phoneNumber, optionalParams);
```