

## Getting Started with

### Getting Started with Voice API

A bit of talking, then three exercises for you to try. Use the time well, we're here to help and we'd love to show you around!

First up: the abbreviation you're going to hear the most today!

### NCCO

### NCCO: Nexmo Call Control Object

### NCCO: Nexmo Call Control Object

Emphasise that these are really important at every stage of voice call applications!

Recordings: start the recording, include a `recording_url`. When complete, webhook goes to `recording_url` with a URL you can download if you are signed in (there are examples in today's exercises).

Audio should be hosted on a public URL, such as S3

DTMF you ask for input, and give the URL to send info to. That URL (synchronously) returns a new NCCO in response to the given data.

Call vs Conferences is the next slide

### Calls vs Conferences

Today's examples only use calls

Use a Conference when you need to know which conversation to have new users join, e.g. conference call or an outgoing call to connect to an existing call when it is answered

### Nexmo Voice API

### Nexmo Voice API

Make the distinction between the NCCO (supplied when the call is started/answered) and the API which operates on the in-progress call and can change things even during the course of an NCCO.

### The Voice API

### NCCO + API = Many Good Things

Combining the NCCOs to control program flow and the API calls to react to events allows us to create interesting and fully-featured applications.

Some things can be done with either technique, but some things need one or the other! Expect to need both.

## **Voice API Examples**

IVR = Interactive Voice Response

Use the proxy to allow customers to connect with staff (for example) without revealing actual numbers

Next: more detail on webhooks

## **Voice Webhooks**

Think of it as a load of pings and beeps to let you know things are happening :)

As well as making API calls, handling webhooks needs your application to receive incoming requests. We'll talk about how to do that on dev platforms in two slides' time

Must ack the webhook, in a specific timeframe. Nexmo will retry if not

Set your URLs through the dashboard or via CLI when you create your application configuration and register a phone number to it

## **Voice Webhooks**

Answer webhook when someone calls your Nexmo number. Nexmo servers hit your URL and you return an NCCO

Events show up at the `event_url`. This is great for knowing what is going on - errors go there too!

DTMF input results in a webhook with a `dtmf` field, you return an NCCO

When recording is ready, webhook to `recording_url` has a link to the file to download

`notify` is an NCCO action that sends a webhook to a URL. The NCCO continues afterwards, useful for progress indicators

There are some tricks to working with webhooks locally: I'd like to share some tips

## **Webhooks on Dev Platforms**

### **Ngrok for Testing Webhooks**

### **The Answer Webhook**

Compare with when we do an outgoing call, we can either specify an `answer_url` and serve an NCCO, or provide the NCCO when making the call

### **The Event Webhook**

### **Voice Events Logger**

### **Pieces of the Voice API**

## Further Reading