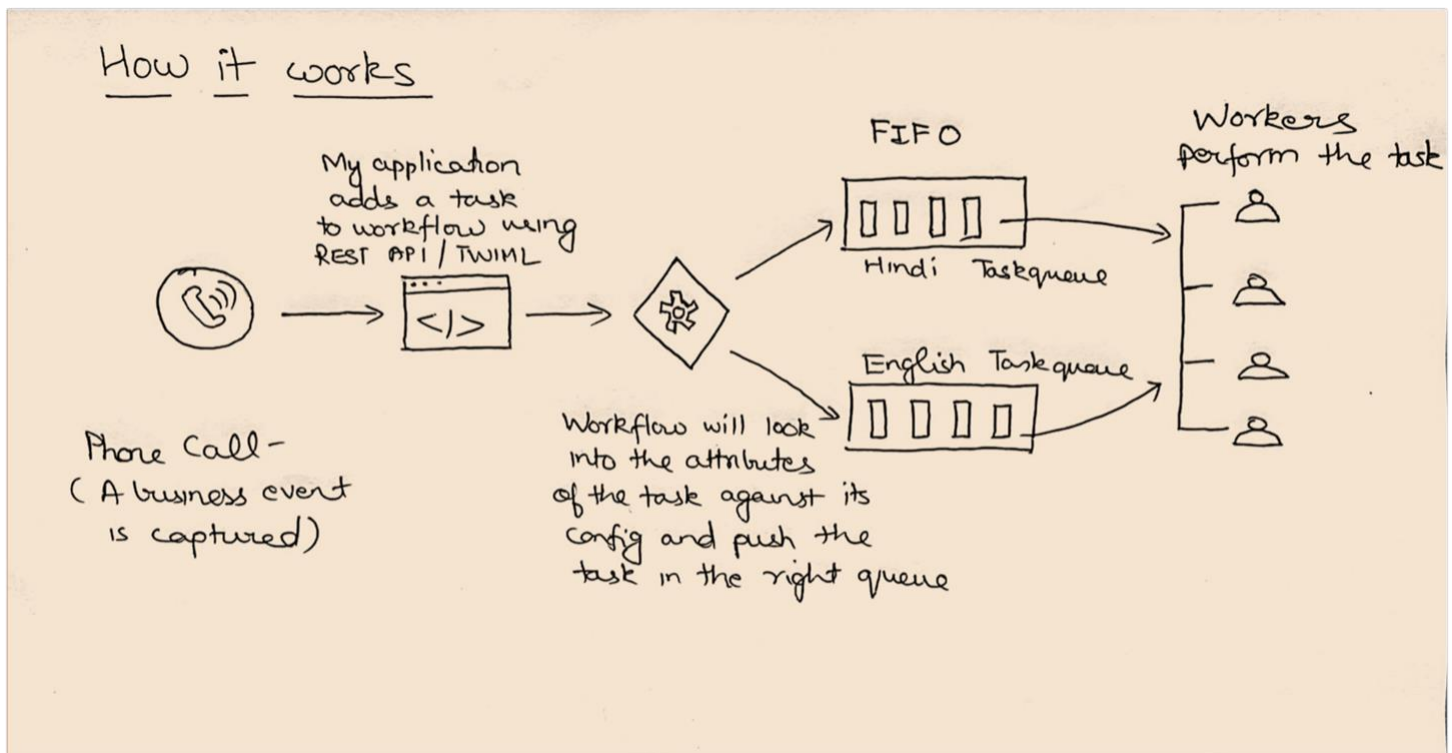


SIMPLE CONTACT CENTRE

TaskRouter is the beating heart of contact centre, made available as an API for developers. Tasks are defined in JSON with a set of customizable attributes such as skills required, priority etc. A phone call is basically creating a task.

The workflow is the brain of the TaskRouter. It inspects the task to understand how to route it based on task attributes assigned at creation. Workflow Objects don't just route and forget but actually monitor tasks as they are in queue. If tasks exceed a defined timeout period without being properly handled, workflow can define how to handle/escalate the task.


How it works



Steps

- 1) Create workspace


Workspaces

 WORKSPACE	DEFAULT ACTIVITY	TIMEOUT ACTIVITY	EVENT CALLBACK URL
Customer Contact Center SID: W5d20d5145ead15e661e3ddf1712f2ff18	Offline	Offline	

2) Create workers and set attributes as appropriate

Shyam Hindi

Properties

WORKER NAME	<input type="text" value="Shyam Hindi"/>
WORKSPACE	WSd20d5145ead15e661e3ddf1712f2ff18
SID	WKbac19f1a5dd92b83d9a083b78ccdb023
ACTIVITY	Offline ▼
ATTRIBUTES 	<pre>{"languages":["hi"],"contact_uri":"+91 8884796044"}</pre>

3) Create task queues

TaskQueues


Search by name or SID

Filter

TASKQUEUE	TOTAL TASKS	MAX RESERVE WORKERS	TARGET WORKER EXPRESSION
<div>Customer Contact Center English</div> <div>SID: W0b4f54ff636bc1d00de74d92dfc8c2eed</div>	0	1	languages HAS "en"
<div>Customer Contact Center Hindi</div> <div>SID: WQ79a3f1e4d5fadd690b271220ef26d4ad</div>	2	1	languages HAS "hi"

4) Create Workflow

Workflows



WORKFLOW	SID	TASK RESERVATION TIMEOUT	ASSIGNMENT CALLBACK URL
Incoming Customer Requests	Ww4687dc7f804994f5de37264538af1325	2 mins	https://liver-rabbit-8458.twil.io/assignment

Assignment Callback

ASSIGNMENT URL https://liver-rabbit-8458.twil.io/assignment

FALLBACK URL

Edit with UI View JSON

Language - English

MATCHING TASKS

selected_language == "en"

Syntax

Routing Steps

MATCHING WORKERS

QUEUE Customer Contact Center English

EXPRESSION Any available worker in queue

ORDERING Longest idle

TIMEOUT & PRIORITY

SKIP TIMEOUT Do not skip

TIMEOUT 1200 seconds

PRIORITY Inherit

5) Set workflow ID and workspace ID in the environment variables

☒ Enable ACCOUNT_SID and AUTH_TOKEN

Environmental Variables

Environmental variables set here will be available to your Functions at runtime.

+

KEY	VALUE	
WORKFLOW_SID	****	
WORKSPACE_SID	****	

When creating custom functions -- uncheck need for valid Twilio signature so that we can view in browser, the TwiML the function generates.

3 parameters which get passed in the function are –

- 1) **context** → contains environment variables which will be used to create Twilio REST client
- 2) **event** → contains details about incoming request
- 3) **callback** → to return the result

```
const client = context.getTwilioClient();
```

Instead of building and scaling out web servers, Twilio functions have been used , a node.js based serverless product designed to make Twilio applications easy to build.

When a user makes a call to Twilio number, Twilio will look up the URL associated with that phone number and sends it a request.

In 'programmable voice' section, under 'manage numbers'- it's been specified to invoke function by name **"Incoming Customer Call"** when a call comes in.

The screenshot shows the Twilio console interface for managing a phone number. The left sidebar contains navigation links: Phone Numbers, Manage Numbers, Active Numbers, Released Numbers, Buy a Number, Verified Caller IDs, Port Requests, Use Your Number, Preview, Addresses, Documents, Tools, Usage, and Getting Started. The main content area is for the number (707) 600-2183, with tabs for Configure, Calls Log, Messages Log, and Event Log. The 'Configure' tab is selected, showing 'Properties' and 'Voice' sections. The 'Properties' section includes fields for Friendly Name, SID, Phone Number, Location, and Capabilities. The 'Voice' section includes a 'Configure With' dropdown, a 'A CALL COMES IN' dropdown menu showing 'Function' selected, and a 'Primary Handler Fails' dropdown menu showing 'Webhook' selected. The 'Call Status Changes' and 'Caller Name Lookup' sections are also visible.

Twilio then reads the TwiML instructions hosted at the URL (/incoming-call POST) to determine what to do.

In this case -playing a welcome message and then prompting the caller to press digits on their keypad.

TwiML **verbs** tell Twilio what *actions* to take on a given call.

<Gather> tag- collect digits the caller types on their keypad

<Say> tag - Reads text to the caller

The action which will run when the digit is entered is 'enqueue-call' (again a POST). Creating a task in queue, waiting for a worker to be available to process the task. Till the worker is unavailable, the user in queue is placed in waiting, with a dummy music playing background. (GET for dummy music)

Once a Worker has been reserved to handle a call, the application receives an HTTP request at the AssignmentCallbackURL specified by the Workflow SID used in the <Enqueue> statement.

Please note- when Twilio performs an HTTP POST to a web server, metadata associated with the call is passed as the body of the POST just as if the parameters were submitted as part of a form on a webpage. Similarly, when Twilio performs an HTTP GET to a web server, metadata associated with the call is passed as URL query string parameters as if the parameters were submitted as part of a form on a webpage.

The basic lifecycle of a [successful] TaskRouter Task is as follows:

Task Created → eligible Worker becomes available → Worker reserved → Reservation accepted → Task assigned to Worker.

So, when a task is created on TaskRouter (passing workflow SID and attributes) its initial state is pending.

Tasks

<div><div><div>+</div></div><div>Priority ▼</div><div>Filter</div></div>							
TASK	PRIORITY	AGE	TASK CHANNEL	TASKQUEUE	WORKFLOW	WORKER	ASSIGNMENT STATUS
WT242736c88235578795b6b48f72728d92	0	1 min 13 secs	voice	Customer Contact Center Hindi	Incoming Customer Requests	Shyam Hindi	assigned
WT401573571ca256f17adad07e50606bd9	0	31 secs	default	Customer Contact Center Hindi	Incoming Customer Requests	—	reserved
WT1c0161373f176d7107a0d34a5815afe6	0	12 secs	default	Customer Contact Center Hindi	Incoming Customer Requests	—	pending

When a worker is selected, following happens in background ---

1. The Task's Assignment Status is set to 'reserved'.
2. A Reservation instance is generated, linking the task to the selected Worker, and TaskRouter starts a configurable task reservation timer (2 mins).
3. At the same time the Reservation is created, a POST request is made to the Workflow's AssignmentCallbackURL.
4. The Reservation has a Status of 'pending' until it is either accepted (confirming the Worker has received the Task), rejected (indicating the Worker explicitly denied the Task), or timed out

Workers are dynamically matched to queues based on skills they have and the skills each queue needs to handle those tasks. Once a worker is available, the highest priority task, longest waiting task will be allocated to the worker. (enqueueing/dequeueing)

A new activity 'Wrap-Up' has been created, which a worker goes into once the task has been completed so that the agent can wrap-up (take notes etc.) before moving on to the next call.

When we dequeue the task, the available worker needs to be called. 'Contact_uri' attribute of the worker is used when routing the call to respective worker.

The 'dequeue' instruction removes the call from hold and bridges it to the selected Worker. **From** parameter specifies the caller ID that should be sent when extending the call to the Worker. **post_work_activity_sid** parameter specifies the ActivitySID that is assigned to the Worker after the call completes.