

## Dialogflow Agent Creation

The screenshot shows the Dialogflow Agent Creation interface for an agent named 'RJTronco-Test-Agent'. The left sidebar contains navigation options: Intents (selected), Entities, Knowledge (beta), Fulfillment, Integrations, Training, Validation, History, Analytics, Prebuilt Agents, Small Talk, and Docs. The main area is titled 'Custom-Flask-intent' and includes a 'SAVE' button. Below the title are sections for 'Contexts', 'Events', and 'Training phrases'. A yellow warning banner states: 'Template phrases are deprecated and will be ignored in training time. More details here.' Below this, a list of training phrases is shown: 'Add user expression', 'supot', 'putso', 'bye', 'python', 'python demo', and 'demo'. The bottom of the sidebar shows 'Trial Free' and an 'Upgrade' button.

## Adding the Flask Webhook

The screenshot shows the 'Fulfillment' section of the Dialogflow interface, specifically the 'Webhook' configuration page. The 'Webhook' toggle is 'ENABLED'. Below the toggle, a text box explains: 'Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.' The 'URL\*' field is filled with 'https://35d3-120-28-248-87.ap.ngrok.io/webhook'. Below this are fields for 'BASIC AUTH' (Enter username, Enter password) and 'HEADERS' (Enter key, Enter value). There is a '+ Add header' button. The 'SMALL TALK' section has a dropdown menu set to 'Disable webhook for Smalltalk'. Below the webhook section is the 'Inline Editor' (Powered by Google Cloud Functions) section, which is 'DISABLED'. A note states: 'Newly created cloud functions now use Node.js 10 as runtime engine. Check [migration guide](#) for more details.' At the bottom, there are tabs for 'index.js' and 'package.json'.

Running NGrok locally to listen to port 5000

```
ngrok
Visit http://localhost:4040/ to inspect, replay, and modify your requests

Session Status      online
Account             Resty John Tronco (Plan: Free)
Version             3.1.0
Region              Asia Pacific (ap)
Latency             109ms
Web Interface       http://127.0.0.1:4040
Forwarding           https://35d3-120-28-248-87.ap.ngrok.io -> http://localhost:5000

Connections          ttl    opn    rt1    rt5    p50    p90
                   6      0      0.00   0.00   0.02   0.02

HTTP Requests
-----
POST /webhook        200 OK
POST /webhook        200 OK
POST /webhook        200 OK
POST /webhook        200 OK
POST /webhook        200 OK
POST /webhook        200 OK
```

Running Flask API

```
test: test
127.0.0.1 - - [07/Oct/2022 12:24:46] "POST /webhook HTTP/1.1" 200 -
rtronco@Restys-MacBook-Air.local /Users/rtronco/Documents/GitHub/Hugging-Face-QnA/ChatBot-Flask-App [main]% flask run
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
demo
haha
127.0.0.1 - - [07/Oct/2022 14:16:30] "POST /webhook HTTP/1.1" 200 -
bye
haha
127.0.0.1 - - [07/Oct/2022 14:17:16] "POST /webhook HTTP/1.1" 200 -
Bye
haha
127.0.0.1 - - [07/Oct/2022 14:30:13] "POST /webhook HTTP/1.1" 200 -
demo
haha
127.0.0.1 - - [07/Oct/2022 17:09:56] "POST /webhook HTTP/1.1" 200 -
bye
haha
127.0.0.1 - - [07/Oct/2022 17:11:00] "POST /webhook HTTP/1.1" 200 -
startbootstrap-grayscale-gh-pages^C
rtronco@Restys-MacBook-Air.local /Users/rtronco/Documents/GitHub/Hugging-Face-QnA/ChatBot-Flask-App [main]: git add
```

## Kommunicate Dashboard when testing

The screenshot displays the Kommunicate Dashboard interface. On the left, a sidebar contains navigation icons. The main area is divided into two sections. The top section, titled 'Assigned', shows a list of chat conversations with their status (Assigned, All, Resolved) and a list of active bots (Ristibot). The bottom section shows a detailed view of a chat conversation with a bot named Ristibot. The conversation history includes messages from the user (Excited Dolphin) and the bot (Ristibot). The bot's responses are: 'Hello! How can I help you?', 'I missed what you said. What was that?', 'Hello from the bot world supot!', and 'Bye ka rin!'. The interface also includes a search bar, a 'Resolve' button, and a 'Take over from Bot' button.

## Web App Integration

The screenshot shows a web browser displaying the Grayscale Bootstrap theme website. The website features a dark background with a mountain landscape and the text 'GRAYSCALE' in large, white, serif font. Below the title, it says 'A free, responsive, one page Bootstrap theme created by Start Bootstrap.' and a 'GET STARTED' button. A chatbot overlay is visible in the bottom right corner, showing a conversation with a bot named Ristibot. The chatbot interface includes a header with the bot's name and status (Online), a list of messages, and a text input field for the user. The messages in the chat are: 'I missed what you said. What was that?', 'Hello from the bot world supot!', and 'Bye ka rin!'. The chatbot is identified as 'Chatbot by Kommunicate.io'.