

# HACK & LEARN



## Mundo dos Bots

# #1 — BOTS MAIS INTELIGENTES

## WEBHOOKS

BANANA  DIGITAL

# WEBHOOKS

Vamos agora deixar nossos BOTS ainda mais inteligentes, finalmente vamos começar a PROGRAMAR.

O que vamos criar é um WEBHOOK. De maneira simplificada, ele é uma maneira de trocar informações entre dois locais na internet.

Nesse caso, vamos comunicar o BOT que criamos no API.AI e um *software* que vamos programar em PYTHON.



api.ai

python

# Codeanywhere

Primeiramente, vamos conhecer a nossa Área de Trabalho, o Codeanywhere.

Ele é uma ferramenta muito poderosa e gratuita que nos permite programar com várias pessoas ao mesmo tempo. (Além de ter uma máquina virtual na núvem!)

Nós já preparamos um espaço para todo o seu grupo programar junto, é só clicar no botão do seu grupo:

**GRUPO 1**

**GRUPO 2**

**GRUPO 3**

**GRUPO 4**

# Codeanywhere

Veja que temos um código chamado:

**“PLAYGROUND”**

Ele tem tudo que você precisa para começar a adicionar funcionalidades ao seu bot.

Dependendo do seu interesse, sinta-se livre para estudar o código. Deixamos ele extremamente documentado.

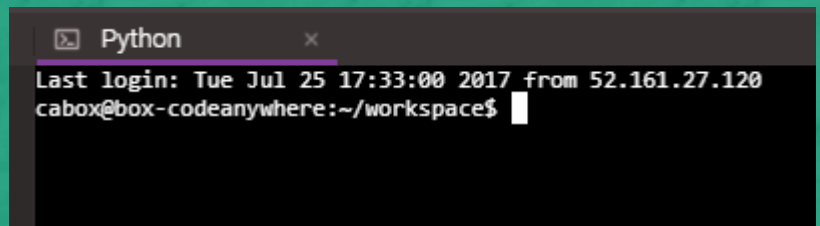
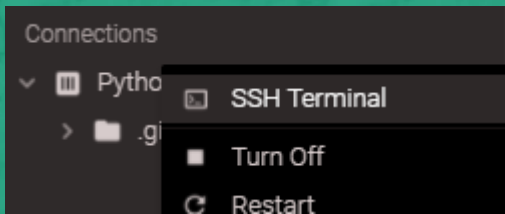
Caso queira ter uma versão sua do PLAYGROUND, temos ele no nosso GITHUB:



(clique na imagem para acessar)

# O Terminal

No CodeAnywhere, fazemos as operações em um Terminal SSH. Isso significa que temos acesso à um computador remoto. No caso o sistema operacional dele é linux Ubuntu.



## Comandos importantes:

`cd nomeDaPasta` (acessar uma pasta)  
`python playground.py` (inicia o webhook)



# Conectar o Webhook

Para usar o Webhook é bem simples.

1- Copie o link fornecido na primeira linha do PLAYGROUND, ele é onde o Codeanywhere está executando nosso projeto.

2- Acesse novamente o seu bot no API.AI e abra a aba de fulfillment:

 Fulfillment

SAVE

## Webhook

Your web service will receive a POST request from API.AI 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](#).

[Webhook example](#)

ENABLED



URL\*

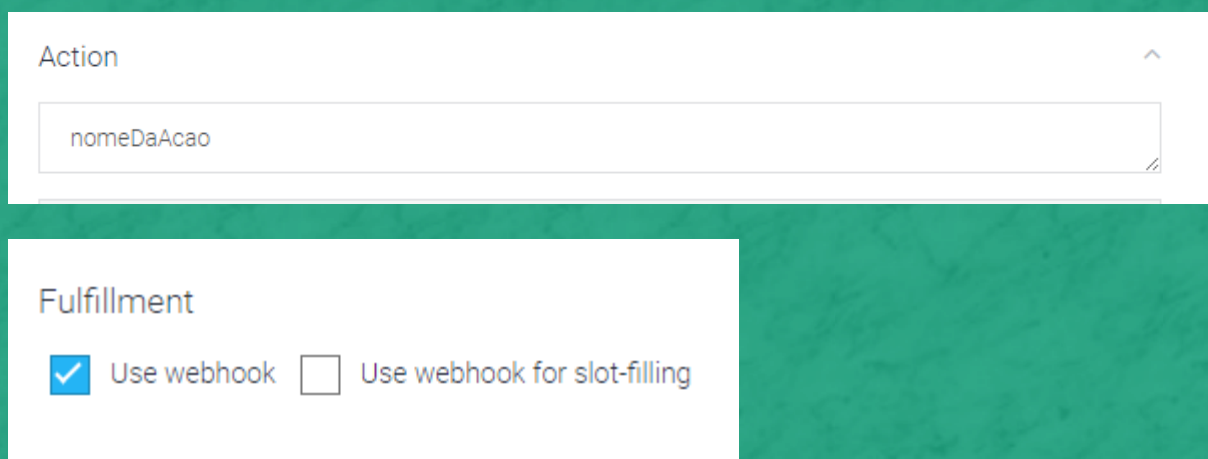
<https://bananadigitalbot.herokuapp.com/webhook>

3- Habilite o Webhook (Enabled)

4- Cole o URL do passo 1

# Como o API.AI chama o Webhook

Se você habilitar o seu intent para usar Webhooks e definir uma ação para ele chamar:



The screenshot shows the configuration interface for an intent action in the API.AI console. It is divided into two main sections: "Action" and "Fulfillment".

**Action:** This section contains a text input field with the placeholder text "nomeDaAcao".

**Fulfillment:** This section contains two radio button options:

- ☒ Use webhook
- ☐ Use webhook for slot-filling

Ele enviará para o Playground os dados no formato JSON descrito no arquivo “API\_REQUEST.json” no seu CodeAnywhere.

O Playground saberá qual ação ele deverá realizar a partir do valor “Action” que você colocou no INTENT.

# Ações no Playground

Na função “processRequest” o seu código irá encontrar qual ação deverá ser realizada a partir de várias comparações:

```
def processRequest(req):  
    action = req.get("result").get("action") #ações  
  
    if action == "#####":  
        #FAZ ALGO  
        retorno = "..."  
        return retorno;
```

Dentro de cada comparação, você pode chamar funções para realizar atividades como mandar e-mails ou fazer consultas.



# Retornos

Todo Webhook deverá retornar algo para o API.AI (nem que seja um valor vazio).

Estas respostas podem incluir os seguintes parâmetros:

Nome	Tipo	Descrição
speech	String	Resposta para a solicitação.
displayText	String	Texto exibido para o usuário.
data	Object	Dados adicionais necessários para a realização da ação no API.AI
contextOut	Array of context objects	Contextos que serão ativados no fim do Intent. Exemplo: <pre>"contextOut": [ {"name": "weather", "lifespan": 2, "parameters": {"city": "Rome"}} ]</pre>
source	String	Fonte dos dados.
followupEvent	Object	Um evento que será chamado após o Intent e parametros adicionais. Permitindo chamar outro Intent logo após a conclusão do Webhook. Exemplo: <pre>{"followupEvent": {"name": "&lt;event_name&gt;", "data": {"&lt;parameter_name&gt;": "&lt;parameter_value&gt;"}}} <a href="#">Read more.</a></pre>

# Exemplo de chamada. (1/3)

Vamos fazer um exemplo, queremos que o BOT retorne o valor da soma de dois números:

” Quanto é 2 + 2

PARAMETER NAME	ENTITY	RESOLVED VALUE	
numero	@sys.number	2	×
numero	@sys.number	2	×

O Intent vai detectar uma lista de parâmetros da entidade número (@sys.number)

Action

somar|

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?	PROMPTS ?
<input checked="" type="checkbox"/>	numero	@sys.numbe r	Snumero	<input checked="" type="checkbox"/>	Define prom pts...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—

E chama a ação “somar” passando os números.

(marcamos o “is list” pois são mais de um valores)

# Exemplo de chamada. (2/3)

Não esqueça de ativar Webhooks no Intent:

Fulfillment

☒ Use webhook ☐ Use webhook for slot-filling


Agora vamos para o nosso código no Playground:

```
def processRequest(req):  
    action = req.get("result").get("action") #ações  
    parameters = req.get("result").get("parameters") #parametros  
  
    if action == "somar":  
        numeros = parameters.get("numeros") #armazenamos os numeros em uma variavel  
        soma = 0 #iniciamos a soma em 0  
  
        for numero in numeros: #para cada valor dos numeros  
            soma = soma + numero #adicionamos ele na soma  
  
        #preparação do retorno  
        retorno = {  
            "speech": soma,  
            "displayText": soma,  
            "source": "app"  
        }  
        return retorno; #envia para o api.ai
```

- 1 - Nos recebemos os valores das ações e parametros enviados.
- 2 - Se a ação for a de somar, coletamos os numeros
- 3 - Inicia a contagem em 0
- 4 - Adicionamos o valor de cada numero
- 5 - Retornamos o valor

# Exemplo de chamada. (3/3)

Podemos rapidamente testar a funcionalidade na Barra Lateral:

calcule  $3 + 5 + 7$ , por favor 


Agent

Domains

USER SAYS

calcule  $3 + 5 + 7$ , por favor

COPY CURL

 DEFAULT RESPONSE

▼

PLAY

15

INTENT

Soma

ACTION

somar

PARAMETER

VALUE

numeros

[3,5,7]

SHOW JSON

# Funcionalidades

Preparamos para você o código de 2 funcionalidades (localizados no seu codeAnywhere):

- Envio de E-Mails
- Solicitação de dados JSON

## E-mail (email.py)

Para mandar um e-mail é muito facil, basta chamar a função:

```
send_email(user, pwd, recipient, subject, body)
```

- user - seu endereço email
- pwd - sua senha
- recipient - endereço email do remetente
- subject - assunto do e-mail
- body - conteúdo do email



# Solicitações de dados JSON (solicitacao.py)

Para fazer a solicitação, você vai precisar de um URL (normalmente o fornecedor dos dados disponibilizará ele)

Exemplo (Retorna a previsão do tempo em Londres):

<http://samples.openweathermap.org/data/2.5/weather?q=London&appid=b1b15e88fa797225412429c1c50c122a1>

Agora só chamar a função `getJSON(url)`

Essa função retornará um objeto JSON que você poderá navegar para encontrar as informações desejadas.

# Navegando nos Dados JSON

Vamos fazer um exemplo com aquele URL de previsão de tempo.

Se abrirmos ele em um visualizador JSON, como [ESSE](#), podemos visualizar a sua estrutura:

<pre>▼ object {1}   ▼ array {12}     ▼ coord {2}       lon : -0.13       lat : 51.51     ▼ weather [1]       ▼ 0 {4}         id : 300         main : Drizzle         description : light intensity drizzle         icon : 09d       base : stations     ▼ main {5}       temp : 280.32       pressure : 1012       humidity : 81       temp_min : 279.15       temp_max : 281.15</pre>	<pre>      temp_max : 281.15       visibility : 10000     ▼ wind {2}       speed : 4.1       deg : 80     ▼ clouds {1}       all : 90       dt : 1485789600     ▼ sys {6}       type : 1       id : 5091       message : 0.0103       country : GB       sunrise : 1485762037       sunset : 1485794875       id : 2643743       name : London       cod : 200</pre>
--	--

Para acessar a temperatura em Londres, o variável seria:

**dados = getJSON(url)**

**Temperatura = dados['main']['temp']**

# Próximo Material:

## #3 — TOQUES FINAIS

BANANA  DIGITAL