**Alexa su Raspberry**

*Lo scopo di questa attività è integrare il sensore e l’attuatore nell’ambiente Alexa. Nel seguito si vedrà come realizzare un semplice skill Alexa, che poi potrà essere arricchito con le funzionalità di controllo del sensore e dell’attuatore.*

*Nella* ***Sezione A*** *si configura lo skill (cioè un contenitore di azioni riconosciute da Alexa) sul sito Amazon; nella* ***Sezione B*** è riportato un esempio di programma associato ad uno skill; nella **Sezione C** si descrive il test dello skill

**Sezione A: configurazione Alexa**

**Parte 1: accesso alla console di sviluppo**

**Fase 1.1:** creare un account di tipo developer, accedendo a

<https://developer.amazon.com/it/>

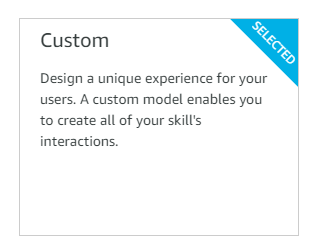
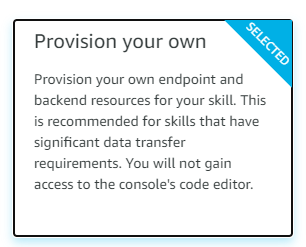
**Fase 1.2:** dal pannello iniziale scegliere ***Alexa***, poi dal menu ***Skill Builder*** scegliere

***Developer Console***

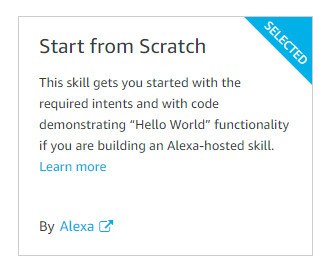
**Parte 2: registrazione dello *skill***

**Fase 2.1:** dalla Developer Console scegliere ***Crea Skill***

**Fase 2.2:** assegnare il nome allo skill (ad esempio *Prova)* e poi selezionare *Custom* e *Provision your own* e scegliere *Create Skill*

*** ***

**Fase 2.3:** scegliere poi *Start from Scratch*e*Continue with template*



**Parte 3: configurazione dello *skill***

**Fase 3.1:** assegnare lo *Skill Invocation Name*  (ad esempio *Prova uno*)*:*  è il nome con il quale si chiederà ad Alexa di attivare lo skill – **salvare il modello**

**Fase 3.2:** nella sezione *Interaction Model*  scegliere *Intents:* gli intents sono le attività che lo skill dovrà eseguire; alcune sono già definite; ad ogni intent corrisponderà una funzione nel programma Python

**Fase 3.3:** creare un nuovo intent, ad esempio *ProvaIntent –* ***salvare il modello***

**Fase 3.4:** assegnare le parole *(Utterance)* che causeranno l’esecuzione della funzione associata all’intent: ad esempio *parla* e *dimmi –* ***salvare il modello***

**Fase 3.5:** esguire il ***Build*** del modello

**Sezione B: sviluppo programma**

**Parte 4: ngrok – rendere visibile il server su Internet**

**Fase 4.1:** registrarsi su <https://ngrok.com/>

**Fase 4.2:** scaricare ***ngrok*** *a* [*questo link*](https://ngrok.com/download) *(versione ARM per Linux)*

**Fase 4.3:** connettere l’applicazione installata all’account seguendo le indicazioni a  
 [questo link](https://dashboard.ngrok.com/get-started/setup)

**Fase 4.4:** consultare[questo link](https://ngrok.com/docs) per l’uso di ngrok e [quest’altro](https://ngrok.com/docs#config_http_proxy) per l’impostazione del proxy

**Parte 5: installare le librerie**

**Fase 5.1:** installare *PIP*

sudo apt install python3-pip

**Fase 5.2:** installare la versione 2.1.4 di *cryptography*

python3 –m pip install cryptography==2.1.4

**Fase 5.3:** installare la versione 0.16.0 di *werkzeug*

python3 –m pip install werkzeug==0.16.0

**Fase 5.4:** installare *flask-ask*

python3 –m pip install flask-ask

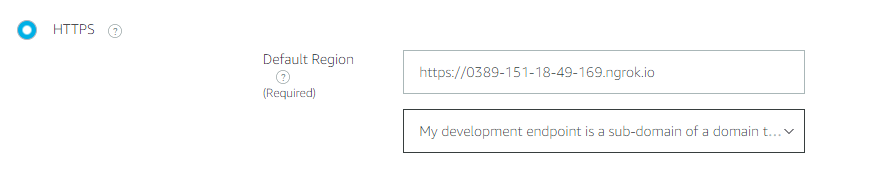
**Parte 6: lanciare il programma di prova**

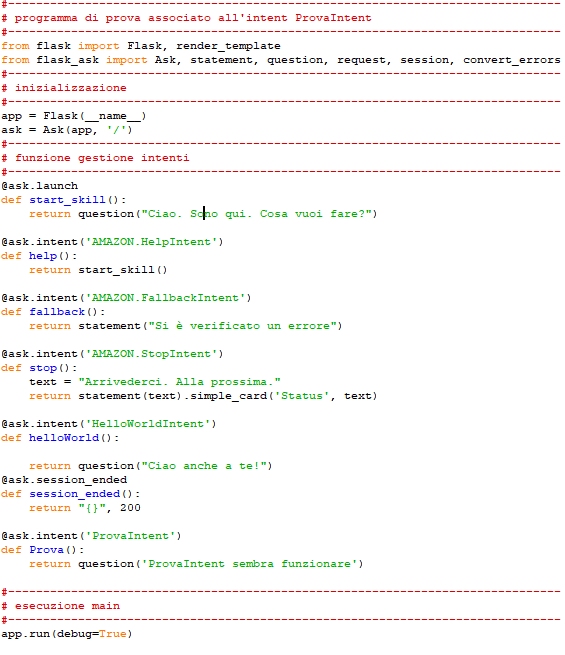
**Fase 6.1:** lanciare il programma alexa.py (riportato in coda in versione copiabile)

**Fase 6.2:** lanciare ngrok e copiare l’url https

./ngrok http 5000

**Fase 6.3:** nella sezione *Endpoint* dello skill di *Developer Console* assegnare l’url fornito da ngrok: le richieste interpretate dal riconoscitore vocale di Alexa verranno inviate a tale url – **salvare l’endpoint**

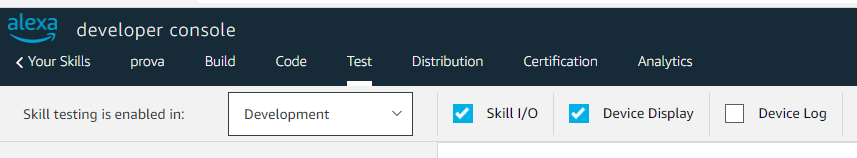
**



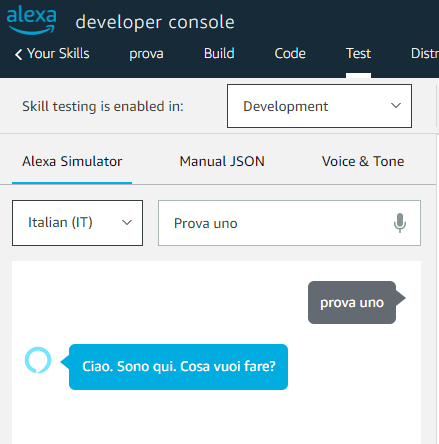
**Sezione C: test**

**Parte 7: provare lo skill dalla Developer Console**

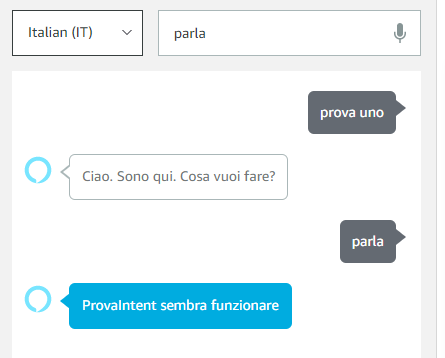
**Fase 7.1:** nella sezione *Test* , abilitare lo skill al test

****

**Fase 7.2:** scrivere o pronunciare lo *Skill Invocation Name*  (ad esempio *Prova uno*)e attendere la risposta prevista nella funzione *start\_skill()*

****

**Fase 7.2:** scrivere o pronunciare uno degli *Utterance* di ProvaIntent(ad esempio *parla*)e attendere la risposta prevista nella funzione *ProvaIntent()*

****

**Parte 8: provare lo skill da Alexa sotto Windows o Android**

**Fase 8.1:** installare Alexa per Windows dallo Store Micorsoft (o da altri siti)o su Android dal Play Store

**Fase 8.2:** loggarsi con l’utenza definita in fase di registrazione (Fase 1.1)

**Fase 8.3:** pronunciare *“Alexa, apri prova uno”*: si dovrebbe ottenere il messaggio di benvenuto definito nella funzione *start\_skill()*

**Allegato – alexa.py**

**#-------------------------------------------------------------------------------**

**# programma di prova associato all'intent ProvaIntent**

**#-------------------------------------------------------------------------------**

**from flask import Flask, render\_template**

**from flask\_ask import Ask, statement, question, request, session, convert\_errors**

**#-------------------------------------------------------------------------------**

**# inizializzazione**

**#-------------------------------------------------------------------------------**

**app = Flask(\_\_name\_\_)**

**ask = Ask(app, '/')**

**#-------------------------------------------------------------------------------**

**# funzione gestione intenti**

**#-------------------------------------------------------------------------------**

**@ask.launch**

**def start\_skill():**

**return question("Ciao. Sono qui. Cosa vuoi fare?")**

**@ask.intent('AMAZON.HelpIntent')**

**def help():**

**return start\_skill()**

**@ask.intent('AMAZON.FallbackIntent')**

**def fallback():**

**return statement("Si è verificato un errore")**

**@ask.intent('AMAZON.StopIntent')**

**def stop():**

**text = "Arrivederci. Alla prossima."**

**return statement(text).simple\_card('Status', text)**

**@ask.intent('HelloWorldIntent')**

**def helloWorld():**

**return question("Ciao anche a te!")**

**@ask.session\_ended**

**def session\_ended():**

**return "{}", 200**

**@ask.intent('ProvaIntent')**

**def Prova():**

**return question('ProvaIntent sembra funzionare')**

**#-------------------------------------------------------------------------------**

**# esecuzione main**

**#-------------------------------------------------------------------------------**

**app.run(debug=True)**