

MegAlexa

Arricchitore di skill di Amazon Alexa

MANUALE DELLO SVILUPPATORE

GRUPPO ZEROSEVEN



Versione	0.0.5
Data Redazione	2019-03-29
Redazione	Mirko Franco
Verifica	??
Approvazione	??
Uso	Esterno
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo ZeroSeven Zero12 s.r.l.
Email di contatto	zerosevenses@gmail.com

Registro delle modifiche

Versione	Data	Descrizione	Autore	Ruolo
0.0.6	2019-04-08	Stesura §4.2.1, §3.1	Stefano Zanatta	Progettista
0.0.5	2019-04-05	Stesura capitolo §4	Mirko Franco	??
0.0.4	2019-04-05	Stesura capitolo §2	Mirko Franco	??
0.0.3	2019-03-30	Stesura capitolo §1	Mirko Franco	??
0.0.2	2019-03-30	Stesura struttura documento	Mirko Franco	??
0.0.1	2019-03-29	Creato documento	Mirko Franco	??

Indice

1	Introduzione	4
1.1	Scopo del documento	4
1.2	Scopo del prodotto	4
1.3	Glossario	4
2	Requisiti di Sistema	5
2.1	Applicazione	5
2.2	Skill	5
3	Procedura di installazione	6
3.1	Installazione Skill	6
4	Tecnologie utilizzate	7
4.1	Amazon Web Service	7
4.1.1	AWS DynamoDB	7
4.1.2	AWS Lambda	7
4.1.3	AWS API Gateway	8
4.1.4	AWS CloudWatch	8
4.1.5	Node.js	8
4.1.6	Npm	8
4.1.7	7z	9
4.2	Dipendenze esterne	9
4.2.1	Alexa Skill	9
4.2.2	App Android	9
5	Architettura	10
6	Estensione delle funzionalità	11

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

1.1 Scopo del documento

Il presente documento vuole essere una guida introduttiva del software *MegAlexa*, indirizzata agli sviluppatori che volessero adattarlo o estenderlo. Vengono spiegate le tecnologie interessate, l'architettura in dettaglio e le possibilità di estensione.

1.2 Scopo del prodotto

Lo scopo del progetto è quello di sviluppare un applicativo Mobile in grado di creare delle routine personalizzate per gli utenti gestibili tramite *Alexa_G* di *Amazon_G*. L'obiettivo è quello di creare *skill_G* in grado di avviare *workflow_G* creati dagli utenti fornendogli dei *connettori_G*.

1.3 Glossario

Al fine di evitare ogni ambiguità di linguaggio e massimizzare la comprensione dei documenti, i termini tecnici, di dominio, gli acronimi e le parole che necessitano di essere chiarite, sono riportate nel *Glossario v3.0.0*. Ogni occorrenza di vocaboli presenti nel Glossario è marcata da una "G" maiuscola in pedice.

Capitolo 2

Requisiti di Sistema

MegAlexa è composta da un'applicazione compatibile con la maggior parte dei dispositivi Android e da una skill Alexa.

2.1 Applicazione

L'applicazione è compatibile con tutti i dispositivi Android con versione 4.4 o superiore.

2.2 Skill

La skill è compatibile con tutti i dispositivi Amazon Echo.

Capitolo 3

Procedura di installazione

3.1 Installazione Skill

Per ognuno dei seguenti comandi è richiesta l'installazione del package manager **npm**. Installazione della skill e delle sue dipendenze:

- clonare la repository attraverso il comando *git clone https://github.com/mirkofranco/ProgettoSWEDocumentazione.git*;
- eseguire il comando *npm install* per installare automaticamente le dipendenze.

Pubblicare la skill in AWS Lambda:

- installare il programma *7z* e inserire il suo eseguibile tra le variabili di sistema;
- installare e configurare *aws-cli*¹;
- da terminal, eseguire il comando *npm run publish-lambda*.

Eseguire i test di unità:

- da terminal, eseguire il comando *npm run unitTest*.

Eseguire i test di integrazione:

- da terminal, eseguire il comando *npm integrationTest*.

¹<https://aws.amazon.com/it/cli/>

Capitolo 4

Tecnologie utilizzate

4.1 Amazon Web Service

Amazon Web Service è una piattaforma di cloud computing sicura che offre servizi di calcolo, memorizzazione, distribuzione di contenuti e altre funzionalità per aiutare il business ad essere scalabile e crescere con facilità. AWS fornisce infatti prodotti e servizi per costruire applicazioni, anche sofisticate, in modo flessibile, scalabile, economico e con un'ottima resistenza ai guasti.

4.1.1 AWS DynamoDB

Amazon DynamoDB è un database che supporta i modelli di dati di tipo documento e di tipo chiave-valore che offre prestazioni di pochi millisecondi a qualsiasi livello. Si tratta di un database multi master, multi regione e completamente gestito che offre sicurezza integrata, backup, ripristino e cache in memoria per applicazioni Internet. DynamoDB può gestire oltre 10 trilioni di richieste al giorno e supporta picchi di oltre 20 milioni di richieste al secondo.

4.1.2 AWS Lambda

AWS Lambda consente di eseguire codice senza dover effettuare il provisioning né gestire il server. Le tariffe sono calcolate in base ai tempi di elaborazione.

Con Lambda, è possibile eseguire codice per qualunque tipo di applicazione o di servizio back-end, senza alcuna amministrazione. Una volta caricato il codice Lambda si prende carico delle azioni necessarie per eseguirlo e ricabarne le risorse con la massima disponibilità. È possibile configurare il

codice in modo che venga attivato automaticamente da altri servizi AWS oppure che venga richiamato direttamente da qualsiasi app Web o mobile.

4.1.3 AWS API Gateway

AWS API Gateway è un servizio completamente gestito che semplifica la creazione, la pubblicazione, la manutenzione e la protezione delle API su larga scala. Con semplicità è possibile creare e configurare API REST che fungano da "porta di ingresso" per le applicazioni, per consentire l'accesso ai dati, alla logica di business o alle funzionalità dai propri servizi back-end. API Gateway gestisce tutte le attività di accettazione ed elaborazione relative a centinaia di migliaia di chiamate ad API simultanee, inclusi gestione del traffico, controllo di accessi e autorizzazioni, monitoraggio e gestione delle versioni delle API. Gateway non prevede alcuna tariffa minima né investimenti iniziali. Vengono addebitati solo i costi di chiamate API ricevute e i volumi di dati trasferiti in uscita e con il modello tariffario a scaglioni di API Gateway potrai ridurre i costi al variare dell'utilizzo delle API.

4.1.4 AWS CloudWatch

AWS CloudWatch è un servizio di monitoraggio e gestione creato per gli sviluppatori, operatori di sistema, ingegneri responsabili del sito e manager IT. CloudWatch fornisce dati e analisi concrete per monitorare le applicazioni, capire e rispondere ai cambiamenti di prestazioni a livello di sistema, ottimizzare l'utilizzo delle risorse e ottenere una visualizzazione unificata dello stato di integrità operativa.

4.1.5 Node.js

La skill è sviluppata attraverso un progetto Node.js. Tutte le classi (appartate `index.js`) sono scritte in Typescript, ma vengono compilate in Javascript prima di essere trasferite in AWS Lambda.

4.1.6 Npm

Gestore di package per node.js. Permette l'esecuzione di comandi personalizzati (come descritto in §3.1).

4.1.7 7z

Permette di comprimere i file da linea di comando. E' richiesto dalla procedura di build della skill.

4.2 Dipendenze esterne

4.2.1 Alexa Skill

Tutte le dipendenze principali si trovano nel file *package.json*, mentre quelle secondarie (cioè le dipendenze delle dipendenze) stanno nel file *package-lock.json*.

Dipendenze principali:

- **ask-sdk:** (Alexa Skill Kit) permette di comunicare direttamente con i servizi Amazon Alexa;
- **axios:** chiamate HTTP. Utilizzato principalmente in *WorkflowService* per le chiamate REST e nei connettori;
- **openweather-apis:** informazioni riguardanti il tempo atmosferico;
- **rss-parser:** trasformazione di un file RSS in un testo più leggibile;
- **twitter:** comunicazione con i servizi Twitter.

Dipendenze utili solo allo sviluppatore (test e compilazione):

- **mocha:** framework per i test in Javascript;
- **chai:** funzionalità aggiuntive mocha;
- **@types/node:** definizioni dei tipi di Node.js per Typescript;
- **@types/mocha:** adattatore mocha per Typescript;
- **@types/chai:** adattatore chai per Typescript;
- **@types/chai-as-promised:** dipendenza aggiuntiva di chai per valutare le *promise*;
- **typescript:** permette la compilazione da Typescript a Javascript.

4.2.2 App Android

Capitolo 5

Architettura

Capitolo 6

Estensione delle funzionalità