Documentation Facebook and Alexa Integration for EVA

Table of Contents

Facebook Integration

Prerequisites Overview

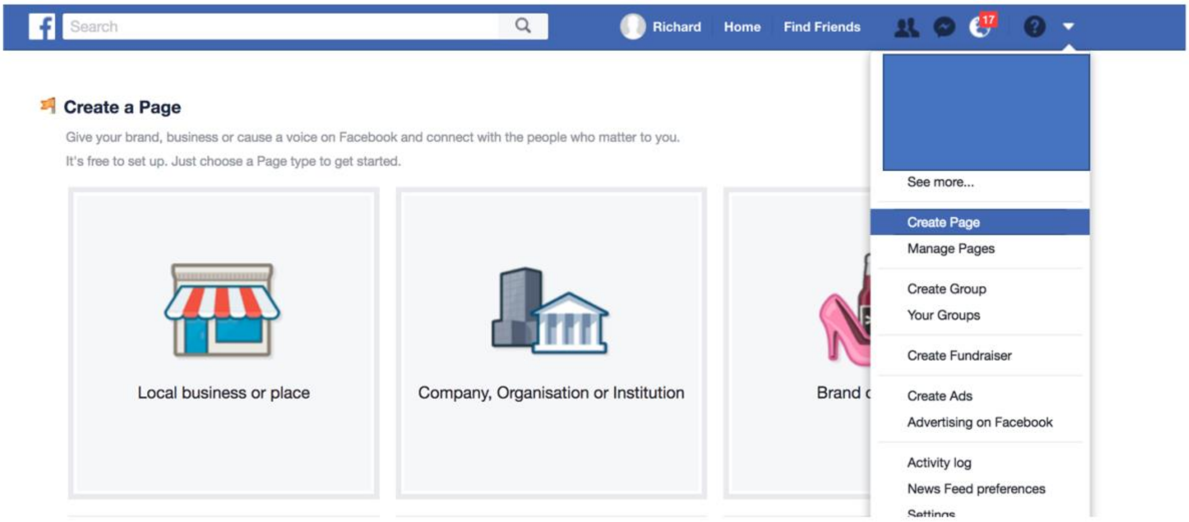
- Facebook (Developer) Account
- Facebook App
- Facebook Feature in EVA
  - Config in DB
  - Add valid client “facebook” in DB
  - Session collection
  - Libraries
    - https://github.com/remixz/messenger-bot/ (MIT License)
- Facebook Test Page

Reference Architecture

TODO

Configuration on Facebook

1. Create Facebook Account on https://www.facebook.com/
2. Register as Facebook Developer on https://developers.facebook.com/
3. Create Facebook Test page

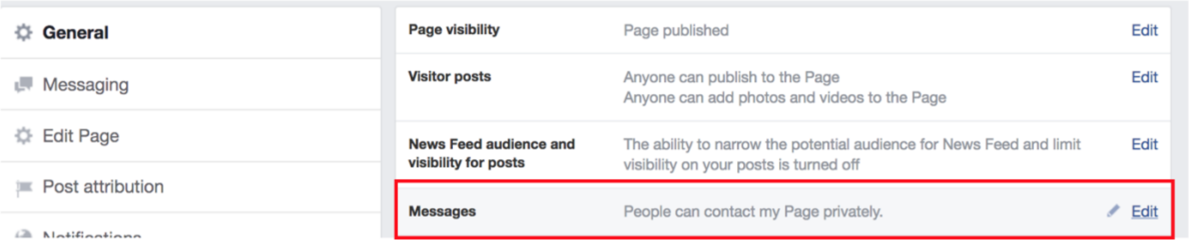


Create a Page

Give your brand, business or cause a voice on Facebook and connect with the people who matter to you.

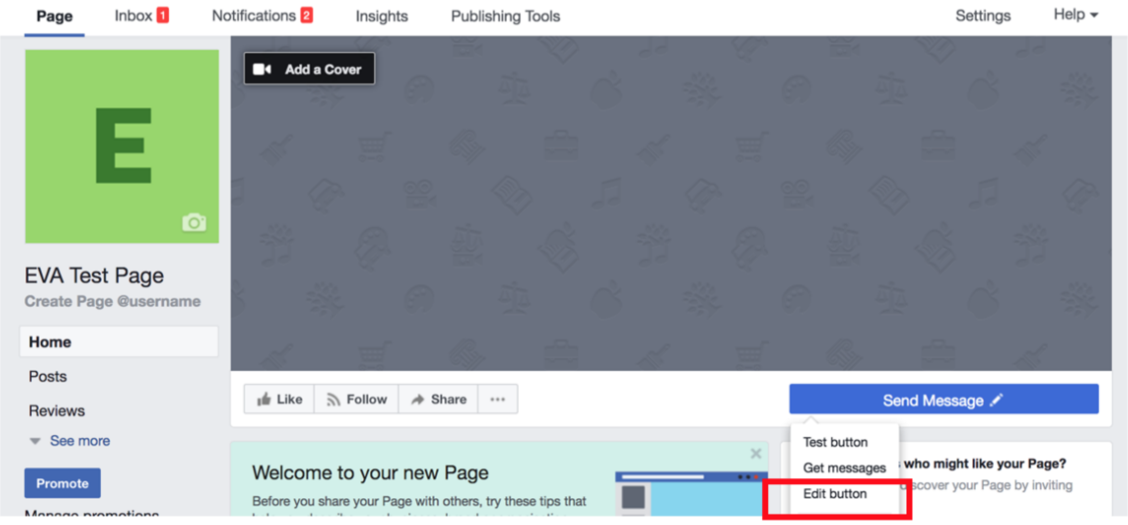
It's free to set up. Just choose a Page type to get started.

4. Ensure that users can send messages to your page privately



Page visibility	Page published	Edit
Visitor posts	Anyone can publish to the Page Anyone can add photos and videos to the Page	Edit
News Feed audience and visibility for posts	The ability to narrow the potential audience for News Feed and limit visibility on your posts is turned off	Edit
Messages	People can contact my Page privately.	Edit

5. Add a "send message" button to your test page



Welcome to your new Page

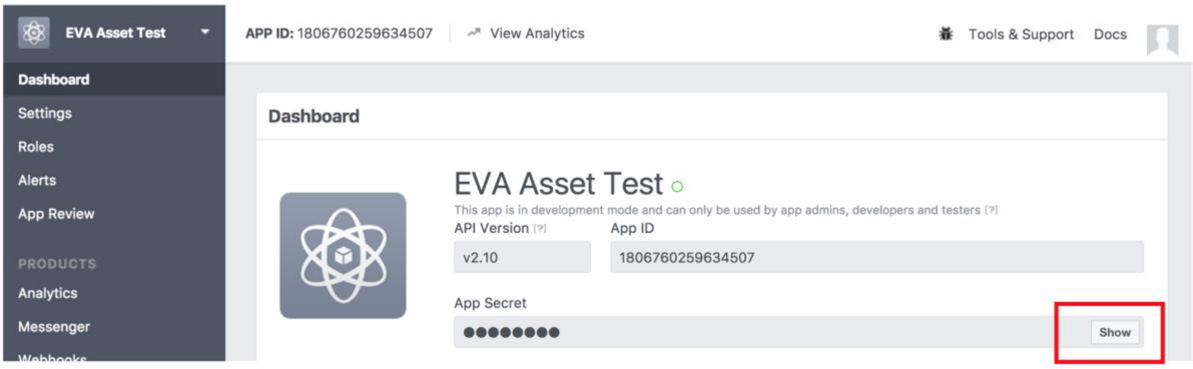
Before you share your Page with others, try these tips that

who might like your Page?

Discover your Page by inviting

Configuration on Facebook Developer Platform

1. Create facebook app on https://developers.facebook.com/
2. Remember the app secret as it is matched to the config field “appSecret” in EVA later



APP ID: 1806760259634507

Dashboard

EVA Asset Test

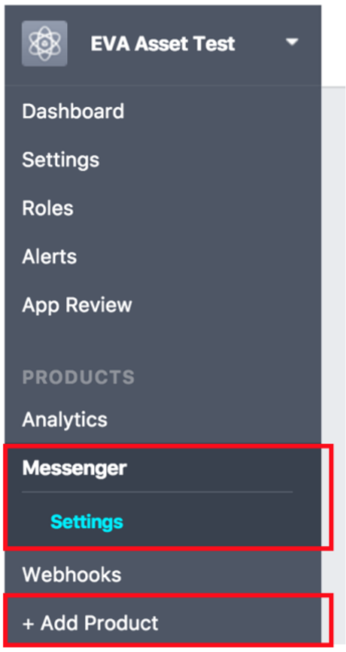
This app is in development mode and can only be used by app admins, developers and testers [?]

API Version [?]: v2.10

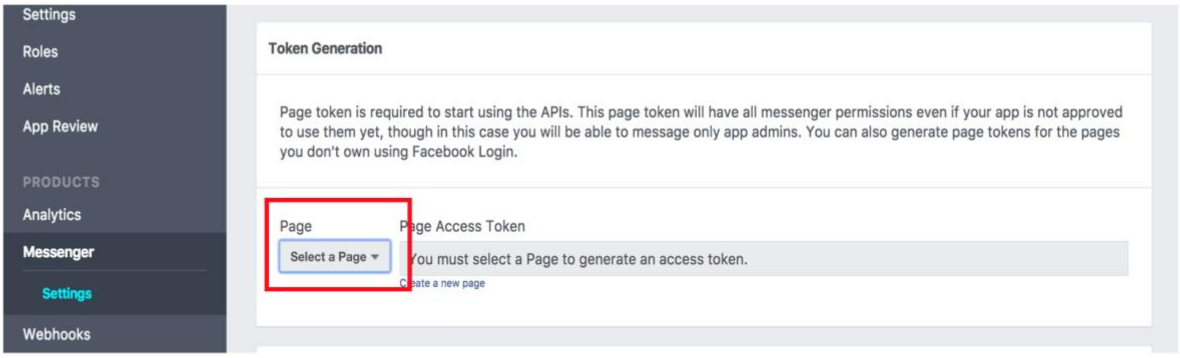
App ID: 1806760259634507

App Secret: ●●●●●●●●

3. Add “messenger” as product to your app



4. Generate Page Token for your prior-created Facebook test page, remember it as it is matched to the config field “token” in EVA later



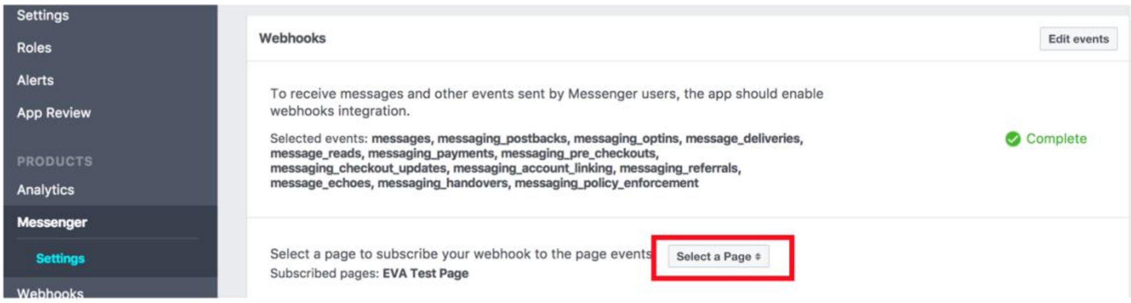
Token Generation

Page token is required to start using the APIs. This page token will have all messenger permissions even if your app is not approved to use them yet, though in this case you will be able to message only app admins. You can also generate page tokens for the pages you don't own using Facebook Login.

Page

Page Access Token: You must select a Page to generate an access token.

5. Subscribe your app or rather its webhook to messaging events on your facebook test page



Webhooks

To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

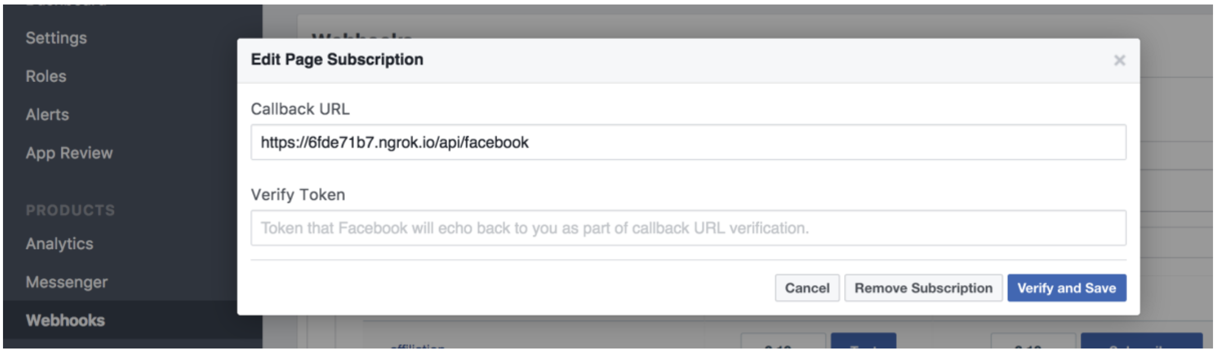
Selected events: **messages, messaging_postbacks, messaging_optins, message_deliveries, message_reads, messaging_payments, messaging_pre_checkouts, messaging_checkout_updates, messaging_account_linking, messaging_referrals, message_echoes, messaging_handovers, messaging_policy_enforcement**

✓ Complete

Select a page to subscribe your webhook to the page events

Subscribed pages: **EVA Test Page**

6. Add “Webhooks” as product to your app in order to change your webhook endpoint (https required!) and verify token, e.g. use ngrok to test your local EVA installation here. Remember the verify token as it is matched to the config field “verify” in EVA later

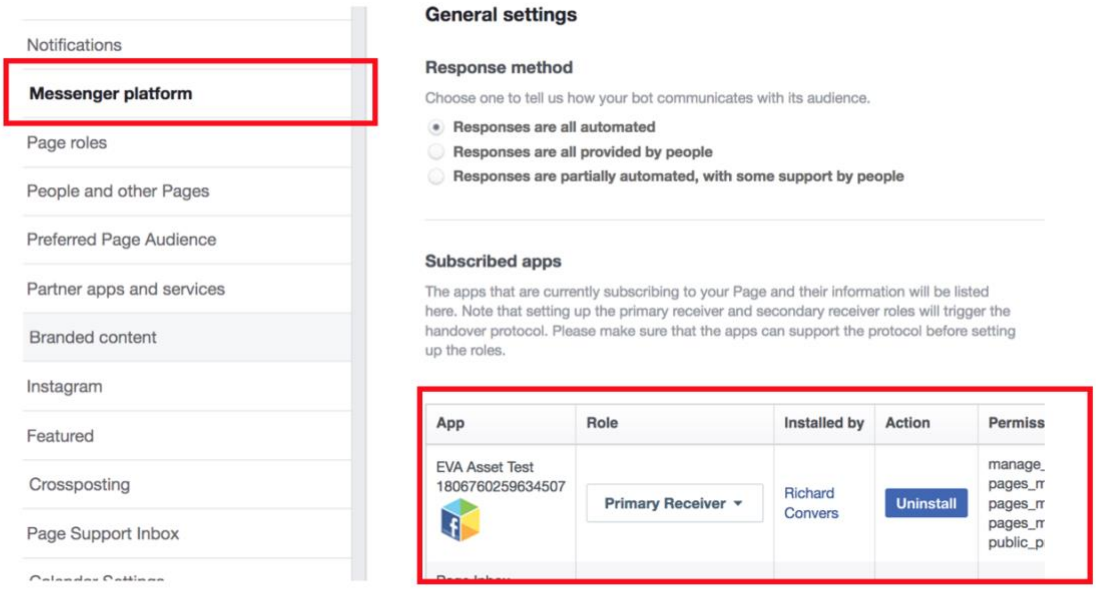


Edit Page Subscription

Callback URL: https://6fde71b7.ngrok.io/api/facebook

Verify Token: Token that Facebook will echo back to you as part of callback URL verification.

7. Ensure that your app is subscribed to your page under your facebook page settings:



General settings

Response method

Choose one to tell us how your bot communicates with its audience.

● Responses are all automated

○ Responses are all provided by people

○ Responses are partially automated, with some support by people

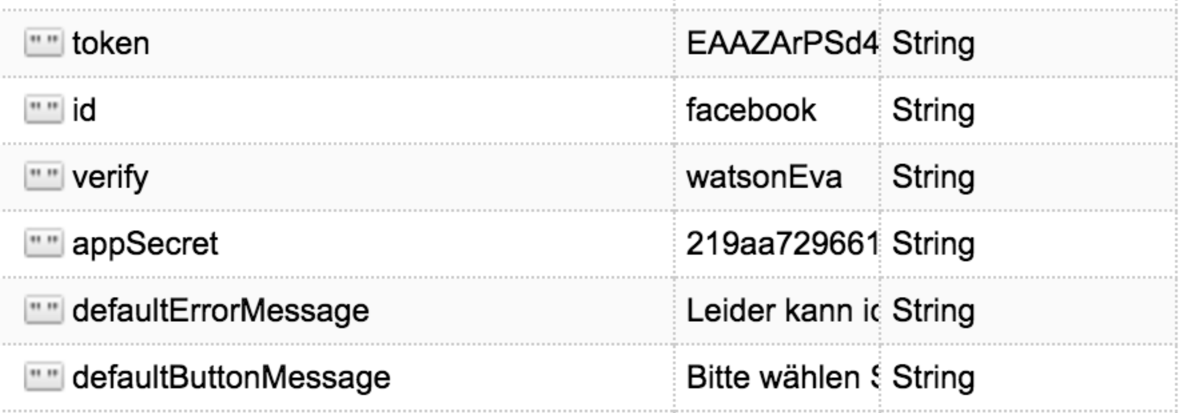
Subscribed apps

The apps that are currently subscribing to your Page and their information will be listed here. Note that setting up the primary receiver and secondary receiver roles will trigger the handover protocol. Please make sure that the apps can support the protocol before setting up the roles.

App	Role	Installed by	Action	Permiss
EVA Asset Test 1806760259634507	Primary Receiver ▾	Richard Convers	Uninstall	manage_ pages_m pages_m pages_m public_p

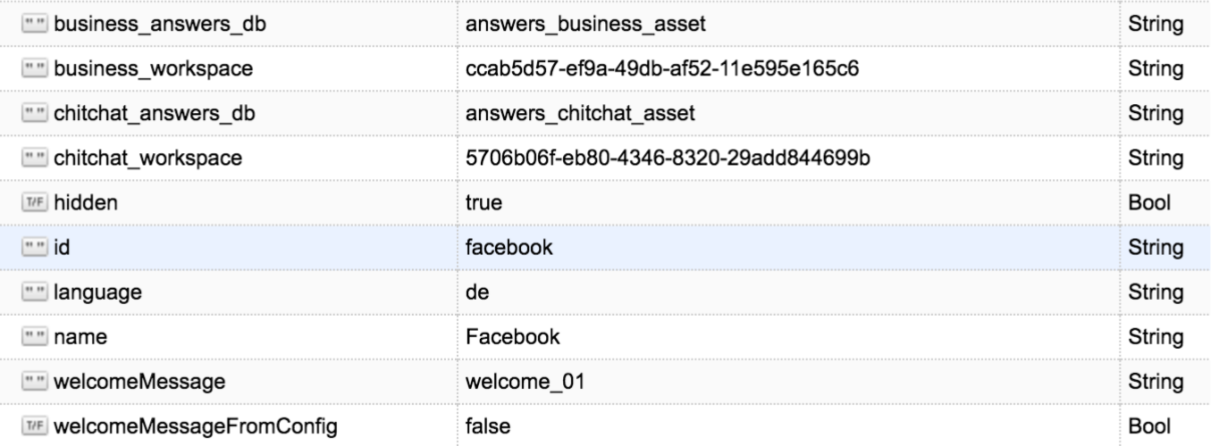
Configuration in EVA

1. Create collection in your DB with name “sessions_facebook”
2. Create a new config in your DB with the following fields as shown in the sample below:



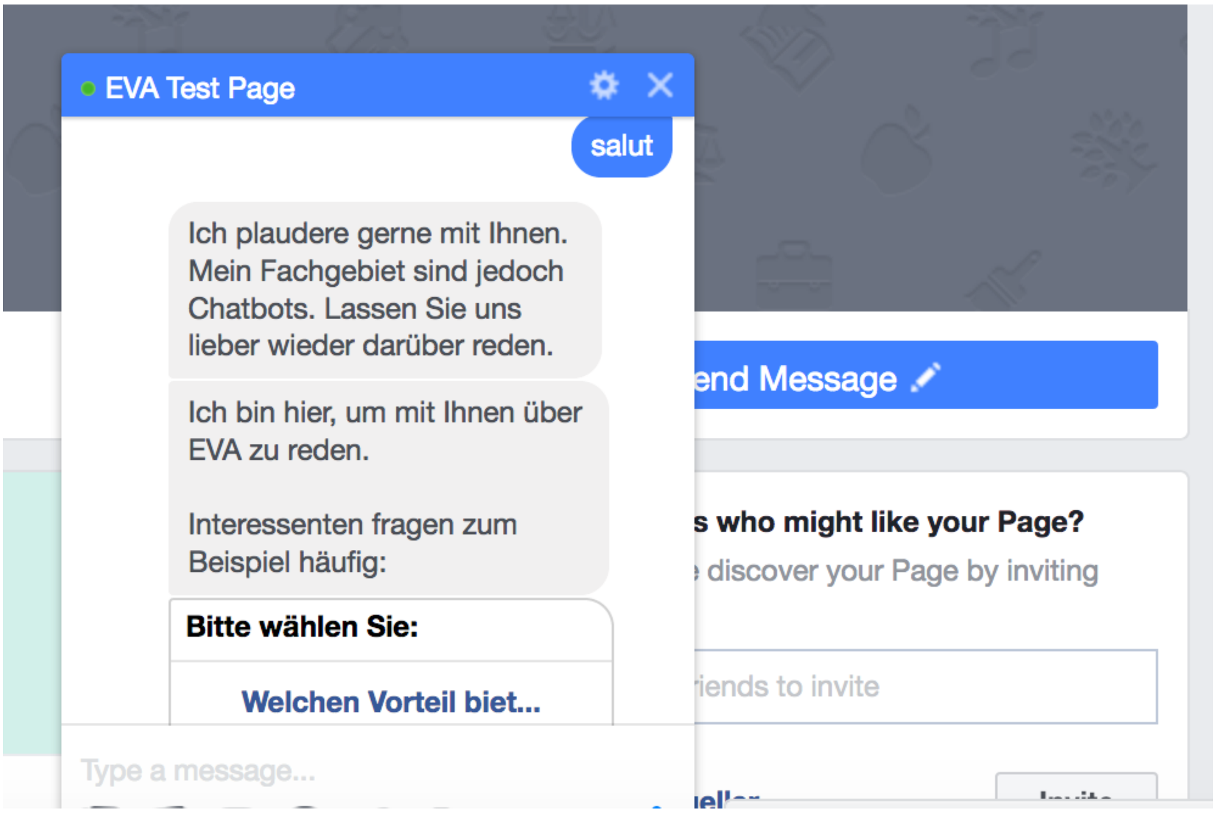
"" token	EAAZArPSd4	String
"" id	facebook	String
"" verify	watsonEva	String
"" appSecret	219aa729661	String
"" defaultErrorMessage	Leider kann ic	String
"" defaultButtonMessage	Bitte wählen ;	String

3. Add new client with id “facebook” to the “clients” collection in your DB



"" business_answers_db	answers_business_asset	String
"" business_workspace	ccab5d57-ef9a-49db-af52-11e595e165c6	String
"" chitchat_answers_db	answers_chitchat_asset	String
"" chitchat_workspace	5706b06f-eb80-4346-8320-29add844699b	String
T/F hidden	true	Bool
"" id	facebook	String
"" language	de	String
"" name	Facebook	String
"" welcomeMessage	welcome_01	String
T/F welcomeMessageFromConfig	false	Bool

4. Start the backend – use e.g. ngrok to test your installation locally. In order to that don’t forget to adjust your webhook setting with the correct endpoint in your facebook app settings as shown before.
5. Go to your facebook test page and test your installation:



salut

Ich plaudere gerne mit Ihnen. Mein Fachgebiet sind jedoch Chatbots. Lassen Sie uns lieber wieder darüber reden.

Ich bin hier, um mit Ihnen über EVA zu reden.

Interessenten fragen zum Beispiel häufig:

Bitte wählen Sie:

Welchen Vorteil biet...

Publish Your App

TODO

Alexa Integration

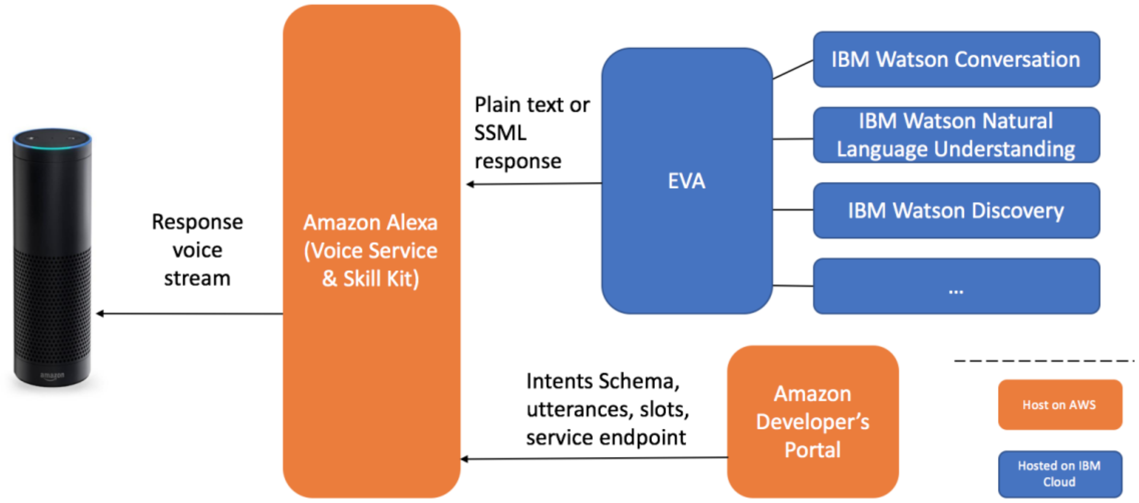
Prerequisites Overview

- Amazon Developer Account
- Alexa Custom Skill on Amazon Alexa platform
  - Invocation Name
  - Intent Schema, Slots, Utterances
  - Service Configuration
- Alexa Feature in EVA
  - Config in DB
  - Add valid client “alexa” in DB
  - Session collection
  - Libraries (recommend if you’re using Amazon Lambda, not required for now)
    - https://github.com/SamVerschueren/aws-lambda-mock-context (MIT License)

        - https://github.com/alexa/alexa-skills-kit-sdk-for-nodejs (Apache 2.0 License)

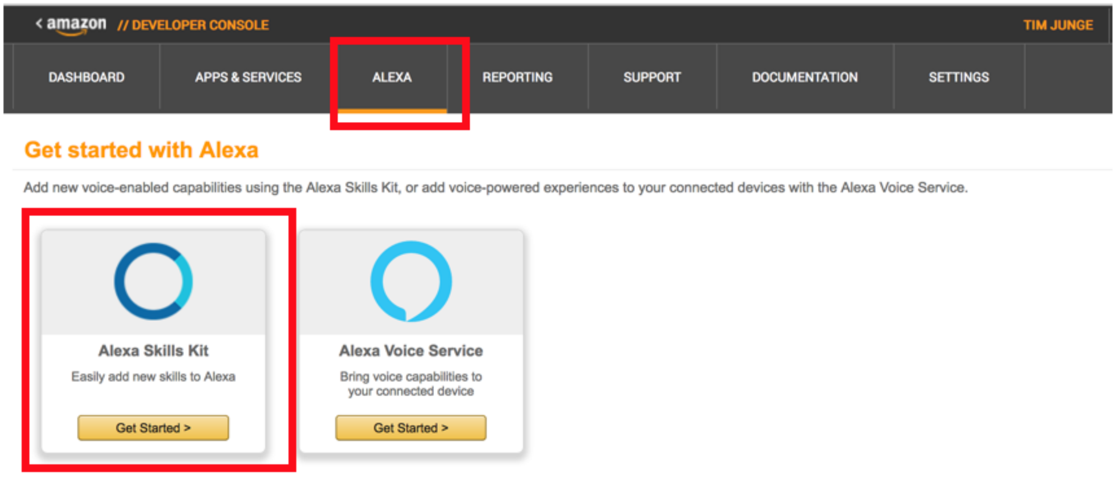
- Testing with device (recommended)
    - Amazon Echo
    - Alexa App on iOS or Android

Reference Architecture


Response voice stream
Amazon Alexa (Voice Service & Skill Kit)
Plain text or SSML response
EVA
IBM Watson Conversation
IBM Watson Natural Language Understanding
IBM Watson Discovery
...
Intents Schema, utterances, slots, service endpoint
Amazon Developer's Portal
Host on AWS
Hosted on IBM Cloud

Configuration on Amazon Developer platform

1. Create Amazon Developer Account on https://developer.amazon.com
2. Got to Alexa Skills Kit



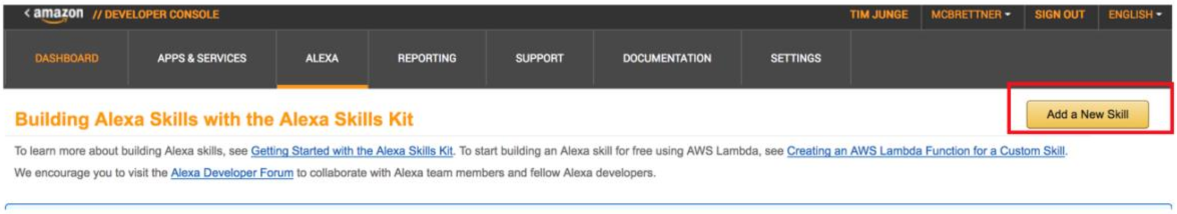
Get started with Alexa

Add new voice-enabled capabilities using the Alexa Skills Kit, or add voice-powered experiences to your connected devices with the Alexa Voice Service.

Alexa Skills Kit: Easily add new skills to Alexa

Alexa Voice Service: Bring voice capabilities to your connected device

3. Add a new Alexa Custom Skill

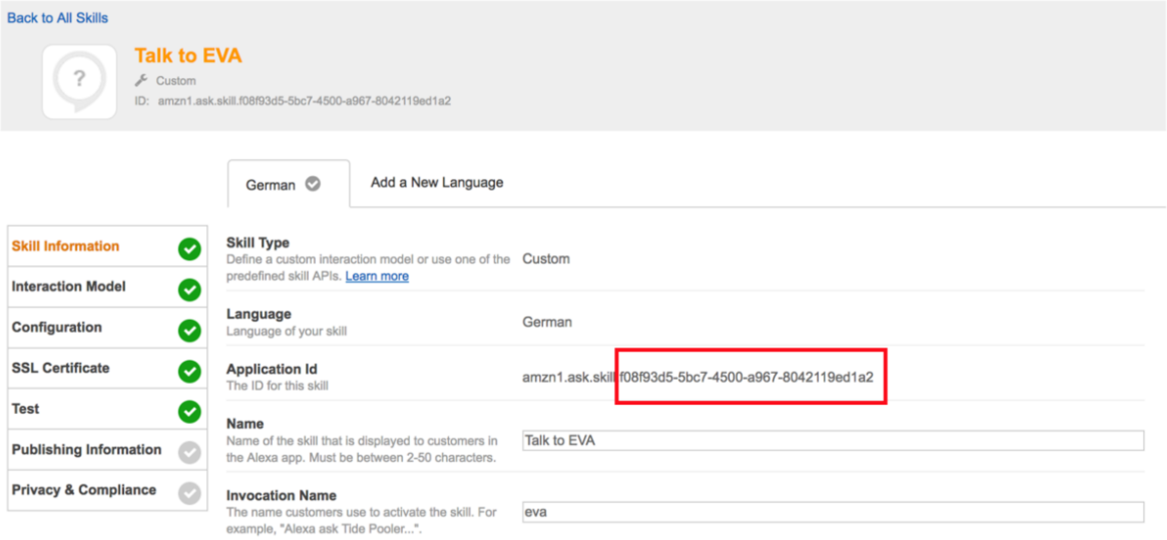


Building Alexa Skills with the Alexa Skills Kit

To learn more about building Alexa skills, see Getting Started with the Alexa Skills Kit. To start building an Alexa skill for free using AWS Lambda, see Creating an AWS Lambda Function for a Custom Skill.

We encourage you to visit the Alexa Developer Forum to collaborate with Alexa team members and fellow Alexa developers.

4. Add required skill information: language, name, invocation name that used to by alexa to identify your skill. Remember the generated skill ID as it is matched to the config field “applicationID” in EVA later



Talk to EVA

Custom

ID: amzn1.ask.skill.f08f93d5-5bc7-4500-a967-8042119ed1a2

Skill Type
Define a custom interaction model or use one of the predefined skill APIs. Learn more
Custom

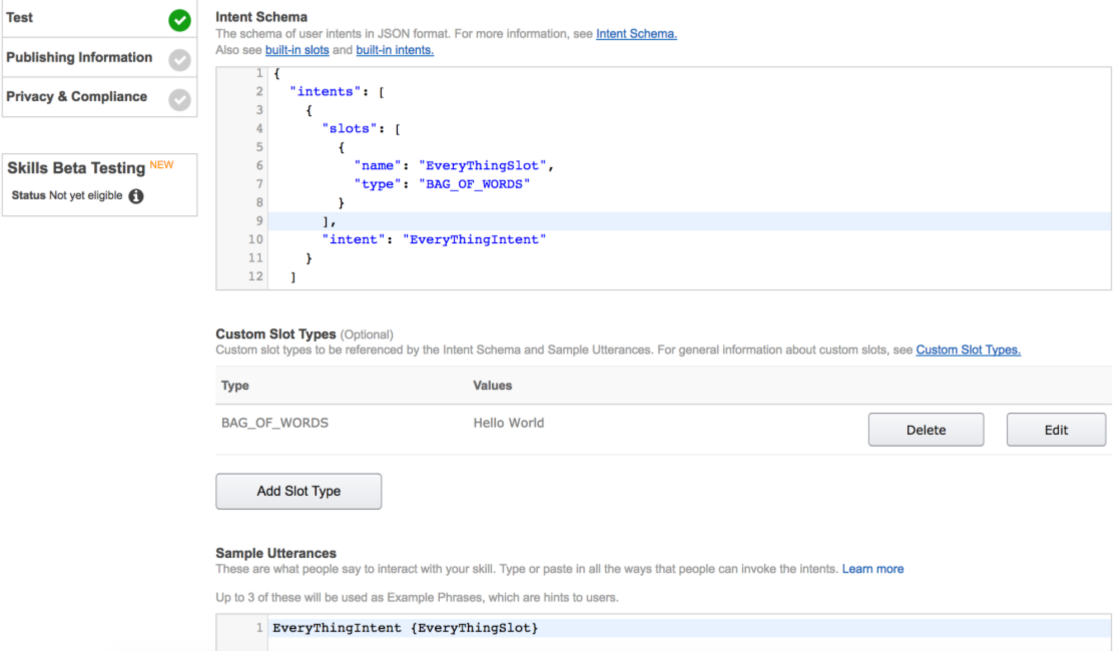
Language
Language of your skill
German

Application Id
The ID for this skill
amzn1.ask.skill.f08f93d5-5bc7-4500-a967-8042119ed1a2

Name
Name of the skill that is displayed to customers in the Alexa app. Must be between 2-50 characters.
Talk to EVA

Invocation Name
The name customers use to activate the skill. For example, "Alexa ask Tide Pooler...".
eva

5. Define the interaction model as shown below:



Intent Schema
The schema of user intents in JSON format. For more information, see Intent Schema.
Also see built-in slots and built-in intents.

```
{
  "intents": [
    {
      "slots": [
        {
          "name": "EveryThingSlot",
          "type": "BAG_OF_WORDS"
        }
      ],
      "intent": "EveryThingIntent"
    }
  ]
```

Custom Slot Types (Optional)
Custom slot types to be referenced by the Intent Schema and Sample Utterances. For general information about custom slots, see Custom Slot Types.

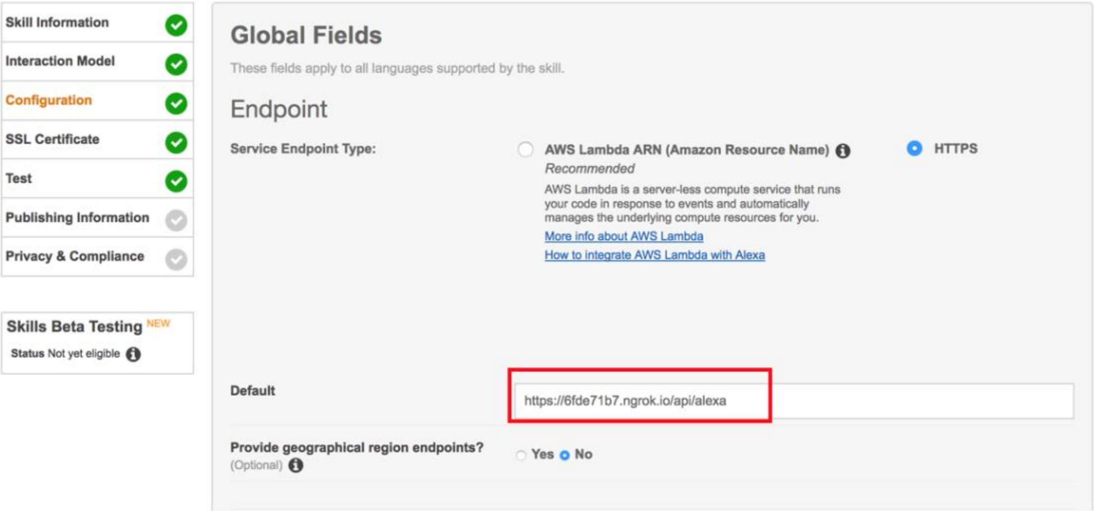
Type	Values
BAG_OF_WORDS	Hello World

Sample Utterances
These are what people say to interact with your skill. Type or paste in all the ways that people can invoke the intents. Learn more

Up to 3 of these will be used as Example Phrases, which are hints to users.

```
EveryThingIntent {EveryThingSlot}
```

6. Configure your service endpoint that is your EVA installation. Use e.g. ngrok to test your local installation, https required



- Skill Information ✓
- Interaction Model ✓
- Configuration ✓
- SSL Certificate ✓
- Test ✓
- Publishing Information
- Privacy & Compliance

Skills Beta Testing NEW

Status Not yet eligible

Global Fields

These fields apply to all languages supported by the skill.

Endpoint

Service Endpoint Type: ○ AWS Lambda ARN (Amazon Resource Name) *Recommended* ● HTTPS

AWS Lambda is a server-less compute service that runs your code in response to events and automatically manages the underlying compute resources for you.

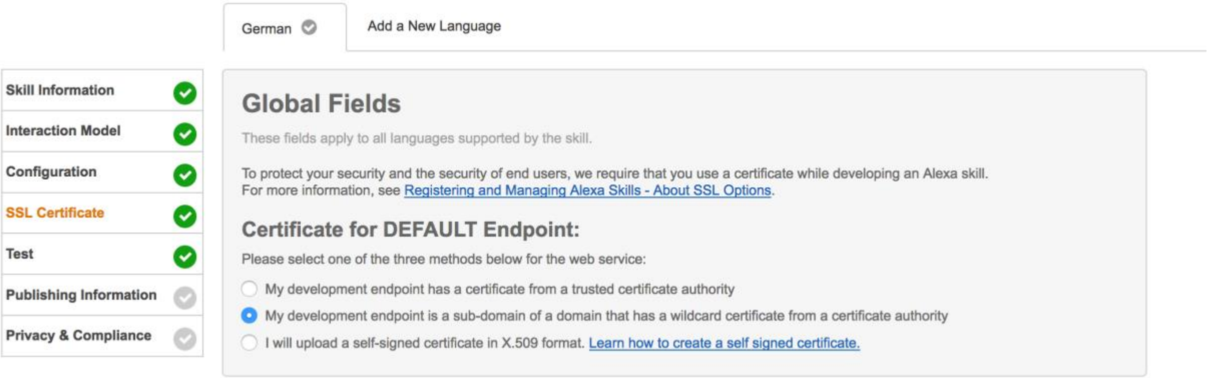
More info about AWS Lambda

How to integrate AWS Lambda with Alexa

Default: https://6fde71b7.ngrok.io/api/alexa

Provide geographical region endpoints? (Optional) ○ Yes ● No

7. Check your ssl certificate for your endpoint as shown below



German ✓ | Add a New Language

- Skill Information ✓
- Interaction Model ✓
- Configuration ✓
- SSL Certificate ✓
- Test ✓
- Publishing Information
- Privacy & Compliance

Global Fields

These fields apply to all languages supported by the skill.

To protect your security and the security of end users, we require that you use a certificate while developing an Alexa skill. For more information, see Registering and Managing Alexa Skills - About SSL Options.

Certificate for DEFAULT Endpoint:

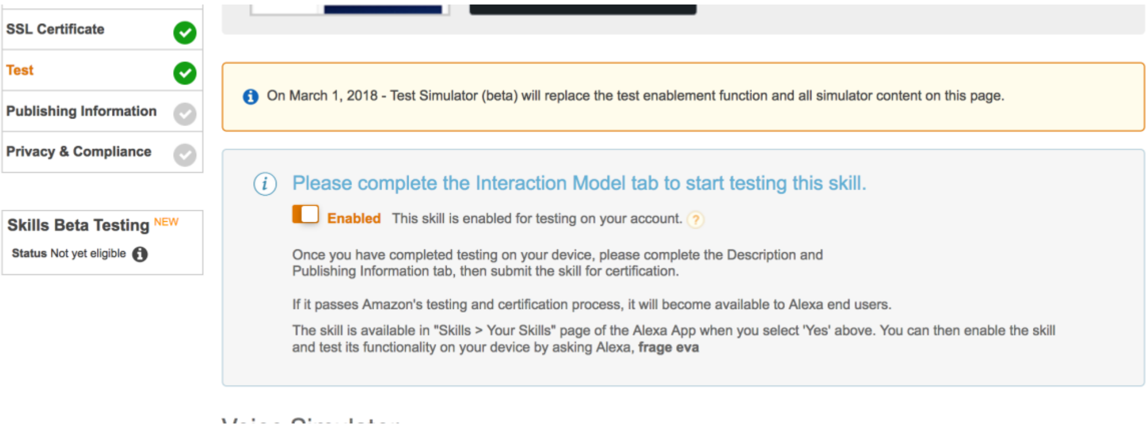
Please select one of the three methods below for the web service:

○ My development endpoint has a certificate from a trusted certificate authority

● My development endpoint is a sub-domain of a domain that has a wildcard certificate from a certificate authority

○ I will upload a self-signed certificate in X.509 format. Learn how to create a self signed certificate.

8. Enable your skill to be tested by your account



- SSL Certificate ✓
- Test ✓
- Publishing Information
- Privacy & Compliance

Skills Beta Testing NEW

Status Not yet eligible

On March 1, 2018 - Test Simulator (beta) will replace the test enablement function and all simulator content on this page.

Please complete the Interaction Model tab to start testing this skill.

Enabled This skill is enabled for testing on your account. ?

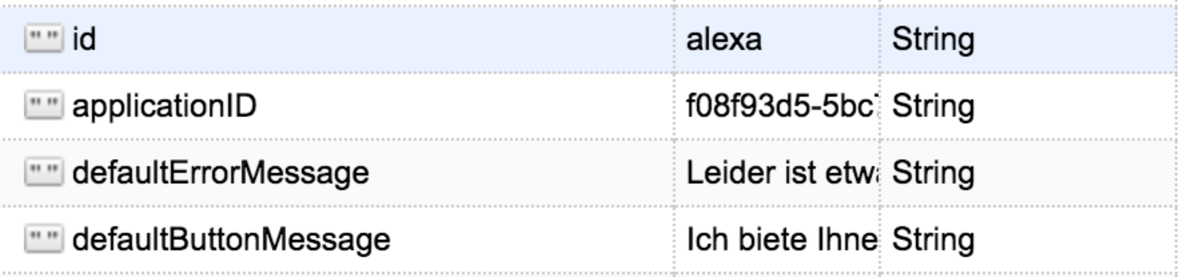
Once you have completed testing on your device, please complete the Description and Publishing Information tab, then submit the skill for certification.

If it passes Amazon's testing and certification process, it will become available to Alexa end users.

The skill is available in "Skills > Your Skills" page of the Alexa App when you select 'Yes' above. You can then enable the skill and test its functionality on your device by asking Alexa, **frage eva**

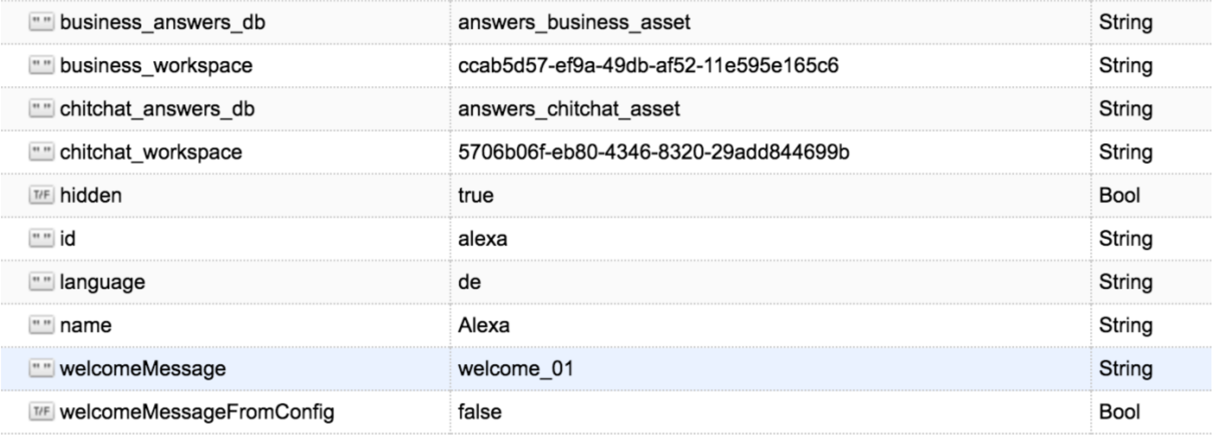
Configuration in EVA

1. Create collection in your DB with name “sessions_alexa”
2. Create a new config in your DB with the following fields as shown in the sample below:



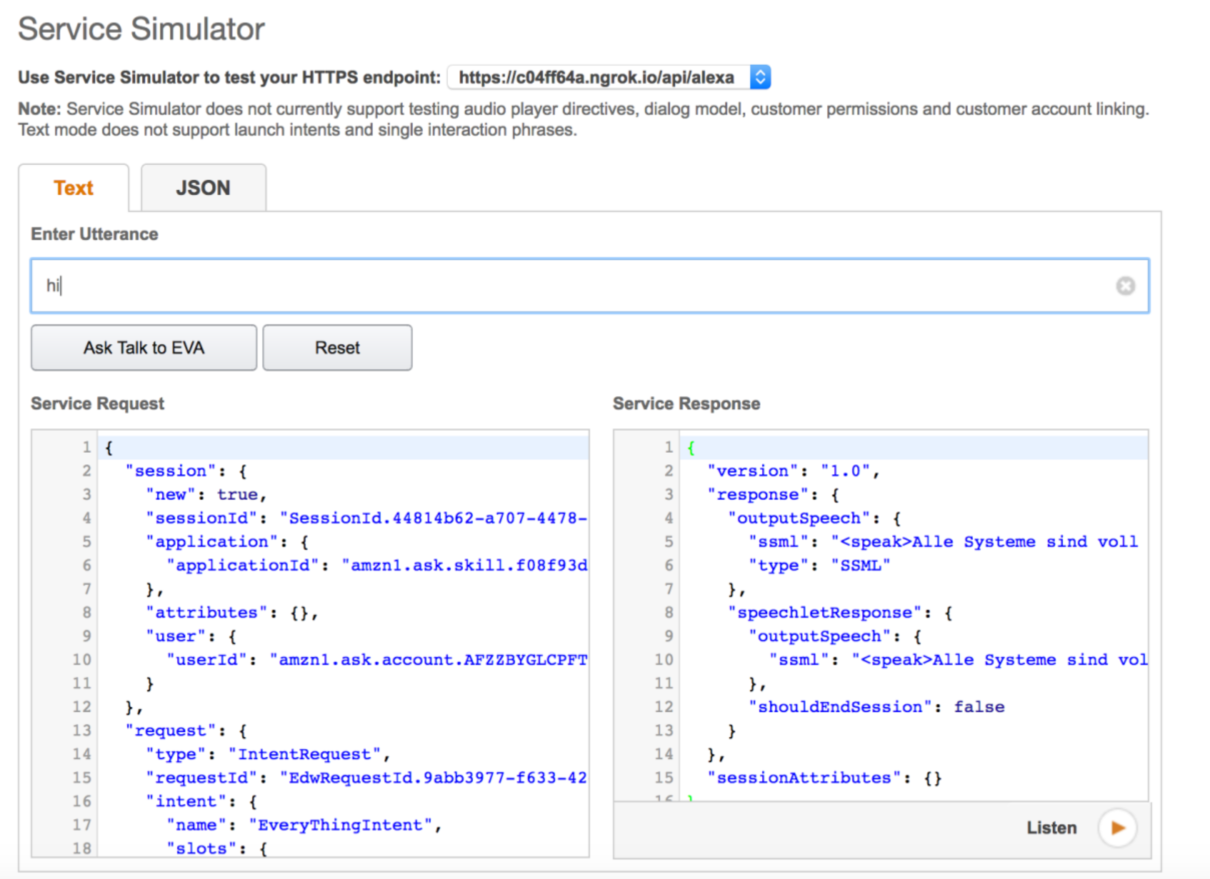
id	alexa	String
applicationID	f08f93d5-5bc	String
defaultErrorMessage	Leider ist etw	String
defaultButtonMessage	Ich biete Ihne	String

3. Add new client with id “alexa” to the “clients” collection in your DB



business_answers_db	answers_business_asset	String
business_workspace	ccab5d57-ef9a-49db-af52-11e595e165c6	String
chitchat_answers_db	answers_chitchat_asset	String
chitchat_workspace	5706b06f-eb80-4346-8320-29add844699b	String
hidden	true	Bool
id	alexa	String
language	de	String
name	Alexa	String
welcomeMessage	welcome_01	String
welcomeMessageFromConfig	false	Bool

4. Start your EVA backend – use e.g. ngrok to test your installation locally. In order to that don’t forget to check your service endpoint settings with the correct endpoint in your alexa skill settings as shown before.
5. Go to your alexa test interface and use the service simulator to test your installation:


Service Simulator
Use Service Simulator to test your HTTPS endpoint: https://c04ff64a.ngrok.io/api/alexa
Text
JSON
Ask Talk to EVA
Reset
Listen

Note: Service Simulator does not currently support testing audio player directives, dialog model, customer permissions and customer account linking. Text mode does not support launch intents and single interaction phrases.

Enter Utterance

hi

Service Request

```
{
  "session": {
    "new": true,
    "sessionId": "SessionId.44814b62-a707-4478-
    "application": {
      "applicationId": "amzn1.ask.skill.f08f93d
    },
    "attributes": {},
    "user": {
      "userId": "amzn1.ask.account.AFZZBYGLCPFT
    }
  },
  "request": {
    "type": "IntentRequest",
    "requestId": "EdwRequestId.9abb3977-f633-42
    "intent": {
      "name": "EveryThingIntent",
      "slots": {
```

Service Response

```
{
  "version": "1.0",
  "response": {
    "outputSpeech": {
      "ssml": "<speak>Alle Systeme sind voll
      "type": "SSML"
    },
    "speechletResponse": {
      "outputSpeech": {
        "ssml": "<speak>Alle Systeme sind vol
      },
      "shouldEndSession": false
    }
  },
  "sessionAttributes": {}
```

Press “Listen” if you want to check how alexa would pronounce your response.

Configuration on Alexa

TODO details

1. Download Alexa App on iOS or Android
2. Log in with your amazon developer account
3. Find and configure your Alexa (echo) device, at least with a valid wifi connection
4. Activate your prior-created custom skill
5. Say “Alexa - Ask invocation name> <your question>

Publish Your App

TODO