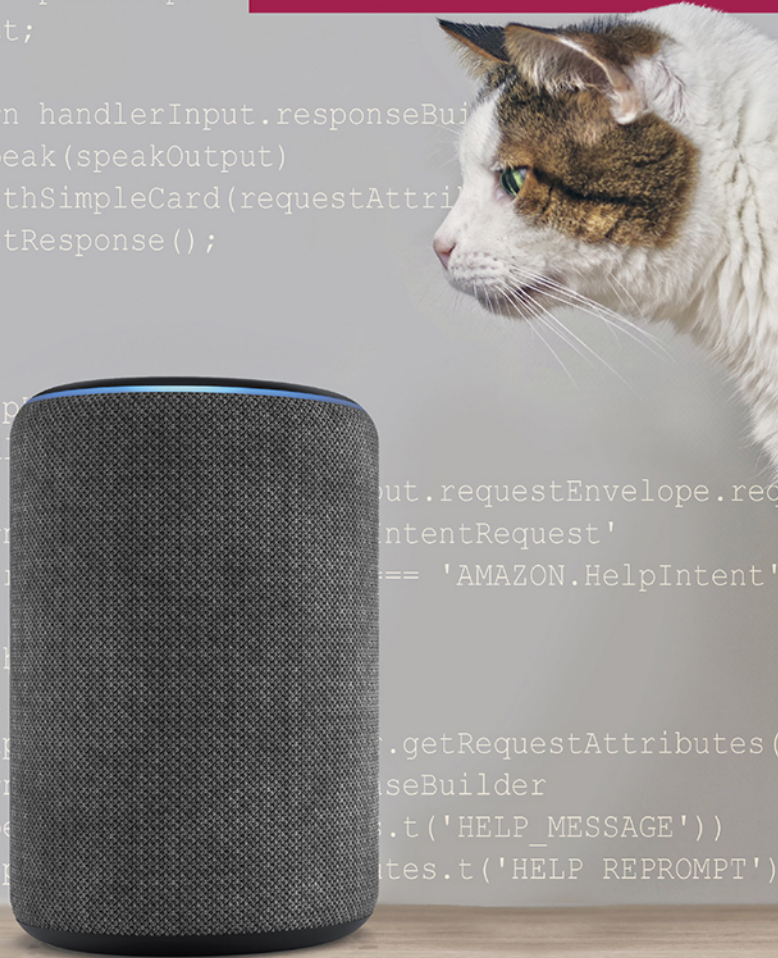




mitp

Alexa Skills programmieren

für Amazon Echo & Co.



Inhaltsverzeichnis

	Einleitung	11
	Warum einen Alexa Skill programmieren?	12
	Über dieses Buch	14
	Über den Autor.	17
Teil I	Grundlagen der Alexa-Skill-Programmierung	21
1	Der Einstieg in die Alexa-Skill-Programmierung	23
1.1	Was ist ein Alexa Skill?	29
1.1.1	Wie wird ein Skill aufgerufen?	30
1.1.2	Wie funktioniert ein Skill genau?	33
1.2	Den Skill in der Alexa Developer Console anlegen	34
1.2.1	Slot-Typ erstellen.	36
1.2.2	Das JSON-Sprachmodell.	40
1.2.3	Alexa-Interface festlegen.	42
1.2.4	Endpoint festlegen	44
1.3	Skill-Logik in Node.js erstellen	44
1.3.1	Node.js-Skill-Code erklärt.	44
1.3.2	Kompletter Skill-Code in Node.js.	52
1.4	Alexa Skill testen mit der Testkonsole	58
1.5	Skill-Code automatisch erzeugen	62
2	Antworten der Skill-Nutzer speichern und verarbeiten	65
2.1	Sitzungsattribute für wiederholte Antworten.	65
2.1.1	Was ist der AMAZON.RepeatIntent?	66
2.1.2	Den AMAZON.RepeatIntent zum Fakten-Skill hinzufügen	66

2.1.3	Dem Sprachmodell einen Intent hinzufügen.	67
2.1.4	Ja/Nein-Absicht mit Sitzungsattributen korrekt erfassen.	73
2.2	Persistente Attribute.	75
2.3	AttributesManager	77
2.4	AWS DynamoDB manuell anlegen.	79
2.5	Debugging mit CloudWatch	82
3	Mit SSML-Tags die Aussprache von Alexa verändern	87
3.1	Flüstern	88
3.2	Audiofiles verwenden.	89
3.3	Pausen	90
3.4	Betonung.	90
3.5	Sprache ändern.	91
3.6	Phoneme	92
3.7	Lautstärke, Tonhöhe und Geschwindigkeit	92
3.8	Satzende	93
3.9	Besondere Betonungen und Aussprache	93
3.10	Wörter ersetzen.	96
3.11	Verschiedene Stimmen	97
3.12	SSML-Tags mit Node.js	98
4	Alexa-Interfaces, Bilder und Videos in Skills einbinden	99
4.1	AWS S3 Buckets	101
4.1.1	S3 Bucket erstellen	101
4.1.2	Ein Bild oder Video in den Bucket laden.	107
4.2	Alexa-Interfaces	111
4.3	Eine Skill Card zu den Antworten erstellen	113
4.3.1	Welche Skill Cards gibt es?	113

4.3.2	Bildformat und Tipps zum Anzeigen von Bildern	118
4.3.3	Wo werden die Bilder abgelegt?	119
4.3.4	Tipps zum Anzeigen des Kartentextes.	120
4.4	Audio-Player	121
4.5	Display-Interface.	122
4.6	VideoApp.	123
4.7	Echo-Buttons.	124
4.8	Custom Interface Controller	125
4.9	Alexa Presentation Language (APL)	126
4.10	Auto Delegation	126

Teil II Fortgeschrittene Methoden und Techniken 127

5	Dialogmanagement mit Alexa	129
5.1	Informationen im Dialog sammeln	129
5.2	Slot-Erfassung mit dem Dialogmanagement.	132
5.3	Das Dialogmanagement einrichten	133
5.4	Nutzerantworten auf Plausibilität prüfen.	138
5.4.1	Marketing by Funny Failing.	140
5.5	Context Switching.	142
5.6	Erinnern und Wiederherstellen des Kontexts mit Sitzungsattributen	145
5.6.1	Der gesamte Code im Überblick	148
5.7	Der Interceptor und seine Bedeutung für das Context Switching.	150
6	Einen Skill mit dem AWS-Cloud-Service erstellen	153
6.1	Einen AWS-Account einrichten	154
6.2	AWS-Lambda-Funktion einrichten und erstellen	159
6.3	Node.js lokal installieren	166

6.4	ASK-CLI Command Line Interface einrichten	172
6.5	Alexa Skill in MS Visual Studio Code programmieren.	184
6.5.1	Schnellstart für das ASK-Toolkit in VS Code	185
7	Externe Informationen im Skill nutzen	195
7.1	Alexa Customer & Settings API nutzen	195
7.2	API Get Request	210
7.2.1	Google Sheets als Content-Backend nutzen . . .	210
7.2.2	Dem Skill einen HTTP-Request hinzufügen.	221
7.3	E-Mail-Versand mit API-Post-Request	231
7.3.1	Mailgun.js installieren	232
7.3.2	Mailgun für die Nutzung in Europa anpassen.	233
7.3.3	Intent-Handler für den E-Mail-Versand	235
7.3.4	Skill in der AWS Cloud und Alexa Developer Console veröffentlichen	246
7.4	Der Post-Request in Node.js	249

Teil III Veröffentlichung und Monetarisierung 253

8	Alexa-Skill-Store-Eintrag erstellen	255
8.1	Beta-Tests einrichten und verwalten	259
8.2	Alexa Skills zertifizieren.	260
9	Alexa-SEO und Skills promoten	269
9.1	Alexa-SEO	269
9.1.1	Was ist ein CanFullfillIntentRequest?	270
9.1.2	Die skalierbare neuronale Architektur hinter Alexas Fähigkeit, Skills auszuwählen	271

9.1.3	Was ist der HypRank?	273
9.1.4	Die Schwierigkeiten bei der Umsetzung	274
9.2	Ihren Skill bekannt machen	278
9.2.1	Designen Sie eine natürliche Launch-Phrase	278
9.2.2	Schreiben Sie eine ausführliche Skill-Beschreibung	281
9.2.3	Nutzen Sie ein auffälliges Skill-Icon	282
9.2.4	Senden Sie eine E-Mail an Ihr persönliches Netzwerk	283
9.2.5	E-Mail-Signatur hinzufügen und Marketing-Symbole aktualisieren	283
9.2.6	Präsentieren Sie Ihren Skill auf Ihrer Website.	284
9.2.7	Nutzen Sie die sozialen Netzwerke zur Skill-Promotion	284
9.3	Marketing-Guidelines für Alexa-Skills	289
10	Geld verdienen mit Alexa.	293
10.1	Account-Linking – Anbindung an einen Onlineshop	294
10.2	Geld verdienen mit Amazon Rewards	298
10.2.1	Wie ermittelt Amazon das Engagement?	302
10.2.2	Was ist eine Interaktion mit dem Skill wert?	305
10.2.3	Wie kann die Interaktionsrate eines Skills verbessert werden?	305
10.3	In-Skill-Purchasing.	311
10.3.1	Welche Arten von In-Skill-Purchasing gibt es?	311
10.3.2	Was motiviert Kunden, einen In-Skill-Kauf zu tätigen?	312
10.3.3	Wie werden In-Skill-Käufe bezahlt?	314

10.4	Einen Skill um Premium-Fakten erweitern	314
10.4.1	In-Skill-Produkte erstellen	317
10.5	Digitale Verbrauchsgüter mit Alexa verkaufen	320
	Rück- und Ausblick	328
	Stichwortverzeichnis	329



Einleitung

Vielen Dank für Ihr Vertrauen und den Kauf dieses Buchs. Die Entwicklung von Amazon Alexa und der Geräte unterliegt einer schnellen Dynamik. Ich habe versucht, die aktuellsten Entwicklungen in diesem Buch nach bestem Wissen und Gewissen zusammenzutragen. Sollte jedoch einer der Punkte bereits überholt sein, wenn Sie dieses Buch lesen, so bitte ich, dies zu entschuldigen.

Was noch vor gut zehn Jahren als Science-Fiction galt, gehört heute zu unserem Alltag. Sprachgesteuerte Geräte halten immer mehr Einzug in unseren Alltag. Fast jeder Smartphone-Nutzer hat die Möglichkeit, mit dem Google Assistant, Bixby oder eben auch Alexa zu interagieren.

Als Alexa-Nutzer und Skill-Entwickler möchte ich Ihnen mit diesem Buch einen Einstieg in die Alexa-Skill-Programmierung mit Node.js an die Hand geben. Ich hoffe, dass ich Sie kurzweilig in die Welt von Alexa einführen kann. Dabei greife ich auf Anekdoten aus meinem Familienalltag und Erfahrungen aus der Alexa-Skill-Programmierung mit der Programmiersprache JavaScript bzw. Node.js zurück.

Im Netz stoße ich immer wieder auf die gleichen Fragen und oft scheint den meisten Menschen nicht klar zu sein, dass es sich bei Alexa um einen cloudbasierten Dienst handelt, der Sprache nur als Interface nutzt. Die Logik dahinter ist meist kein Hexenwerk und hat nur indirekt etwas mit künstlicher Intelligenz zu tun, je nach Definitionsart mehr oder weniger.

Trotzdem neigen wir Menschen dazu, Alexa im Alltag zu vermenschlichen. Das liegt vor allem an der Art und Weise, wie wir Alexa nutzen. In der Geschichte gab es noch nie eine Maschine, die wir einfach per Sprache bedienen konnten.

Als Alexa-Skill-Programmierer schwindet diese Illusion. Ein Ziel dieses Buchs ist es, Alexa und andere sprachgesteuerte Systeme zu entmystifizieren. Ich möchte Sie dazu ermächtigen, selbst den Ton in Ihren eigenen vier Wänden wieder anzugeben.

Wenn ich es dann noch schaffe, Sie dazu zu inspirieren, eigene Skills zu erstellen, hat dieses Buch seinen Zweck vollends erfüllt.

Also genug der Vorworte, packen wir es an!

Warum einen Alexa Skill programmieren?

Skills erweitern die Möglichkeiten von Alexa. Als Skill-Betreiber können Sie über das Alexa-Sprachinterface in Kontakt mit Ihren Nutzern treten und mit ihnen interagieren.

Als Suchmaschinenoptimierer und Publisher diverser Nischenblogs bekomme ich die Diskussion über den Wandel Googles von einer Suchmaschine zur Antwortmaschine mit. Viele SEOs kämpfen nun um »Position 0«. Das ist die Sprachantwort zu einer Suchanfrage und aktuell das oberste Ergebnis zu vielen Suchanfragen in Google.

Diesen Zirkus möchte ich als Publisher nicht mehr mitmachen, denn der Nutzen für den Publisher ist nicht erwiesen, wenn Google hergeht, sich den Content einer Seite schnappt und dem Nutzer die Antwort selbst sagt. In den meisten Fällen gelangt dann der Nutzer nicht auf die Seite des Publishers, sondern beendet seine Suche.

Als Publisher ist man der Dumme, wenn man hier nur auf Google vertraut.

Der Trend zur Voice Search ist aber ungebrochen, so sollen im Jahr 2020 über 50% aller Suchen per Stimme erfolgen. Was können Sie also machen, um trotzdem Ihre Nutzer zu erreichen, ohne dass Sie von Google dabei übervorteilt werden?

Für mich war die Antwort schnell klar, als ich 2014 von Alexa hörte, dass Amazon sein System offen für Entwickler an den Markt bringen wollte. Ich wollte von nun an meinen Content primär über Alexa Skills veröffentlichen.

Zudem ist Sprache das natürlichste Interface, über das wir Menschen mit unserer Umwelt interagieren. Schon im Mutterbauch kommen wir mit Sprache in Berührung.

Sprachgesteuerte Systeme, wie Alexa mit ihren Skills, setzen Nutzungsbarrieren sehr niedrig an. Schon Kleinkinder können mit Alexa interagieren und ihre Ziele erreichen. Ältere Menschen haben oft keine Lust, neue Bedienkonzepte zu lernen, aber kaum eine Hemmschwelle, das Interface zu nutzen, das sie von Kindesbeinen an kennen: die Sprache.

Halten wir also fest, warum Alexa Skills in naher Zukunft sehr wichtig sein werden:

- Die Verbreitung von Alexa als Sprachassistent ist sehr groß und wächst von Jahr zu Jahr dramatisch. Allein über 11 Millionen Nutzer in Deutschland nutzen einen Sprachassistenten aktiv und suchen nach neuen Skills.
- Sprache wird Tasten- und Wisch-Interfaces ersetzen, weil man mit ihr schneller komplexe Sachverhalte kommunizieren kann.
- Wir können mit Sprache und ihren unterschiedlichen Formulierungen eine Absicht beschreiben und verschiedenste Teilaspekte nebenbei betonen. Ein Web-Text ist schlicht zu starr, um ihn auf verschiedenste Absichten auszurichten. Eine Lösung mit vielen ähnlichen Web-Texten läuft aber dem Optimierungs-Credo von Google entgegen. Mit einer Voice-App bzw. einem Alexa Skill kann man ganz flexibel auf unterschiedliche Sprachmuster passend eingehen.
- Komplexe Tasks werden seit Neuestem von Alexa mit unterschiedlichen Skills in Kombination erledigt.
- In der Suchmaschinenoptimierung gib es den Trend weg von Keyword und Suchphrasen hin zur Optimierung von Webseiten nach Intents (Absichten). Nur ist eine Website eine Krücke im Vergleich zu einer Voice-App bzw. einem Skill, wenn es um die Bedienung eines Intents geht. Wenn Sie eine Website in Google optimieren möchten, müssen Sie bestimmte Regeln einhalten, um auch gut gefunden zu werden. Leider stehen diese Regeln oft einer Optimierung nach einem bestimmten Intent entgegen.

- Das Suchen mit Keywords und Suchphrasen war gestern. Es gab schon immer eine Absicht, die hinter einer Recherche steckte, diese wird jetzt direkt mit Sprachassistenten wie Alexa befriedigt.
- Sie können mit einem Skill noch relativ einfach generische Themenfelder besetzen.
- Sie können mit Ihrem Skill Geld verdienen und mit sogenannten »In-Skill Purchases« kostenpflichtigen Premium-Content anbieten.

Über dieses Buch

Dieses Buch richtet sich an alle, die in die Alexa-Skill-Programmierung einsteigen möchten. Grundkenntnisse in JavaScript oder Webprogrammierung sind von Vorteil und erleichtern das Verständnis dieses Buchs. Ich habe jedoch darauf geachtet, Ihnen gleich am Anfang auch eine Möglichkeit zu zeigen, wie JavaScript-Code auch mithilfe eines Code-Generators im Netz automatisch erzeugt werden kann. So sind selbst blutige Anfänger in der Lage, einen ersten Skill zu erstellen.

Dieses Buch möchte vor allem anderen motivieren, in die Alexa-Skill-Entwicklung einzusteigen. Sie werden Schritt für Schritt in die Programmierung und in das Skill-Design eingeführt.

Aufbau des Buchs

Dieses Buch ist grob in drei Teile gegliedert. **Teil I** (Kapitel 1-4) beschäftigt sich mit den Grundlagen der Alexa-Skill-Programmierung.

In **Kapitel 1** erkläre ich den Einstieg in die Alexa-Skill-Programmierung. Hier begleite ich Sie durch die Erstellung eines Alexa-Developer-Accounts und erkläre die grundlegende Funktionsweise eines Alexa Skills.

In **Kapitel 2** zeige ich Ihnen, wie Sie den Kontext für Ja- und Nein-Antworten des Nutzers mit einem Skill erfassen können. Zudem erkläre ich den Unterschied zwischen Sitzungsattributen und persistenten Attributen.

Kapitel 3 befasst sich mit der Aussprache von Alexa und wie Sie diese mit SSML-Tags beeinflussen können.

Kapitel 4 stellt die Alexa-Interfaces in den Mittelpunkt. Es werden hier also Ausgabemöglichkeiten vorgestellt, die Sie potenziell mit einem Skill nutzen können. Außerdem wird hier die Frage geklärt, wie Sie Bilder und Videos für Ihren Skill verwenden können.

In **Teil II** (Kapitel 5-7) gehe ich auf die fortgeschrittenen Methoden und Techniken in der Skill-Programmierung ein.

Kapitel 5 widmet sich dem Dialogmanagement. Hier erkläre ich, wie Sie Zug um Zug Informationen im Dialog von einem Nutzer sammeln können und so aus den Antworten des Nutzers eine Antwort von Alexa erstellen können. Für Fortgeschrittene ist dann das Thema Kontextwechsel interessant, also die Frage, wie ein Skill damit umgeht, wenn ein Nutzer mal etwas anderes zwischenfragt, sodass er im Dialog nicht wieder von vorne anfangen muss.

In **Kapitel 6** erkläre ich, wie Sie einen Skill in der AWS Cloud erstellen und Ihren Skill mit Microsoft Visual Studio Code programmieren können.

Kapitel 7 beschäftigt sich mit dem Thema API und wie Sie externe Informationen für Ihren Skill nutzen können. Sie lernen hier, wie Sie eine einfache externe Datenbank mit Hilfe von Google Sheets aufbauen und in Ihrem Skill für die Antwortausgabe nutzen können – oder, was für viele Onlinemarketer interessant sein dürfte, wie Sie die gegebenen Nutzerantworten per Post-Request in eine eigene Datenbank speichern bzw. für den E-Mail-Versand nutzen können.

In **Teil III** (Kapitel 8-10) werden schließlich die Skill-Promotion, die Vermarktung des Skills und Alexa-SEO thematisiert.

In **Kapitel 8** erkläre ich, wie Sie Ihren Skill im Alexa Skill-Store eintragen und gebe Ihnen Tipps, wie Sie Ihren Skill von Amazon zertifiziert bekommen.

Das **Kapitel 9** behandelt das Thema Alexa-SEO und erläutert Ihnen die Grundlagen, wie Alexa einen bestimmten Skill vorschlägt. Zudem gebe ich hier Tipps, wie Sie ihren Skill im Netz bekannt machen können.

In **Kapitel 10** gehe ich auf die Thematik Voice Commerce ein und zeige Wege auf, wie Sie als Unternehmer bzw. Publisher mit Alexa Geld verdienen können.

Downloads zum Buch

Alle Codebeispiele finden Sie zum Download auf der Website des Verlags unter www.mitp.de/0114.

Wer Interesse an weiteren Code-Beispielen hat, dem empfehle ich auch das Repository von Amazon Alexa selbst. Sie finden es unter <https://github.com/alexa> oder, wenn Sie Interesse an den neuesten Entwicklungen haben, unter <https://github.com/alexa-labs>.

Über die Projekte im Buch

Das Skill-Beispiel mit den Weltraumfakten, das im Verlauf des Buchs erstellt wird, ist ein rein theoretisches Projekt und wurde noch nicht veröffentlicht. Es wurde nur zu Anschauungszwecken erstellt und getestet. Darüber hinaus nutze ich in diesem Buch aber auch Code von Skills, die schon so in der freien Wildbahn und von Amazon zertifiziert sind. Hier werden mitunter nur einzelne Teilaspekte zur Veranschaulichung genutzt.

Was nicht in diesem Buch steht

Als Autor dieses Buchs fiel mir die Entscheidung recht schwer, welche Inhalte ich ausführlich darstellen und welche Themen ich nur anreißen sollte. An dieser Stelle möchte ich mich bei meiner Lektorin Janina Bahlmann bedanken, die mir immer wieder Hinweise gab, was für einen Leser interessant sein könnte und was für Anfänger eher schwer verständlich ist.

Das Thema APL, also Alexa Presentation Language, habe ich in dem Buch nur kurz angerissen. Es ist derart umfangreich, dass man hierzu schon fast ein eigenes Buch schreiben könnte. Zudem finden in diesem Teilaspekt der Skill-Programmierung immer wieder Änderungen und Neuerungen statt. Auch das Thema Skill-Zertifizierung ist dabei

nochmal etwas komplexer, da hierbei auch die Bildschirmausgabe mitberücksichtigt wird.

Dann gibt es noch das Thema Amazon Pay, also die Möglichkeit, per Sprache einen eigenen Onlineshop anzubinden und unabhängig vom Amazon Marketplace reale Waren per Sprache zu verkaufen. Für mich persönlich ist das eines der spannendsten und von vielen noch unterschätzten Themen, aber eben auch für den Einstieg leider zu umfangreich und komplex.

Auch das Thema Account Linking wird hier im Buch nur sehr kurz angerissen. Die Idee hierbei ist, dass der Nutzer seine Profildaten von einem anderen Internetdienst nutzen kann und seine Skill-Nutzung in einer externen Datenbank dauerhaft gespeichert und genutzt werden kann.

All diese Themen wollte ich Anfängern oder Anfängerinnen nicht zumuten. Das wäre die Aufgabe eines anderen, umfangreicheren Buchs.

Über den Autor



Sammy Zimmermanns ist Marketing-Kommunikationswirt (DAMK), seit 2006 ist er Blogger und Suchmaschinenoptimierer und berät seine Kunden, wie sie in der Suchmaschine Google nach vorne kommen und mehr Nutzer erreichen.

Er hat die Entwicklung des Onlinemarketings seit vielen Jahren verfolgt und online diverse Fachartikel zum Thema Suchmaschinenoptimierung und Alexa veröffentlicht.

Sammy Zimmermanns hat auf Fachkonferenzen zu dem Themen WordPress, Suchmaschinenoptimierung und Alexa-Skill-Programmierung gesprochen und betreibt den Alexa-Stammtisch Dresden. 2018 wurde er als Top SEO Influencer vom SEO-Portal ausgezeichnet. Seit 2016 beschäftigt er sich intensiv mit der Alexa-Skill-Programmierung in der Sprache Node.js.

Er lebt mit seiner Frau und seinen beiden Kindern in Dresden. In seiner Freizeit treibt er gerne Kraft- und Ausdauersport. Sie erreichen ihn per E-Mail unter: info@internet-pr-beratung.de.

Wie ich zum Programmierer wurde

Ich bin eigentlich kein Programmierer von Hause aus, wer ist das schon? Denn eigentlich bin ich Blogger über Kaffee und SEO-Themen und berate meine Kunden, wie sie bei Google möglichst weit vorne in der Suche erscheinen. Alle Kenntnisse habe ich mir autodidaktisch beigebracht – mithilfe des Internets und den Vorlagen, die Amazon allen Entwicklern frei zur Verfügung stellt.

Ich habe einen Echo Dot seit Herbst 2016 und gehöre zu dem Kreis der Nutzer, die nur auf Einladung einen Amazon Echo bestellen konnten. Die ganze Entwicklung hat mich fasziniert und ich war seit der Vorstellung von Alexa gefesselt von dem Gedanken, hier selbst etwas zur Verbesserung dieser Sprachassistentin beitragen zu können.

Erste Berührung mit der Skill-Programmierung hatte ich durch ein Video auf YouTube des YouTubers Venix (<https://www.youtube.com/watch?v=v7vVywojhzE>). Dort wurde Schritt für Schritt erklärt, wie man einen Google-Suche-Skill für Alexa erstellt.

Damals fand ich noch, dass eine Google-Suche für Alexa eine prima Sache wäre. Ich habe mich also an den Rechner gesetzt und bin dieses Tutorial Schritt für Schritt durchgegangen. Nach ein wenig Code-Frickelei war ich stolzer Ersteller eines Google-Suche-Skills für meine Alexa.

Neugierig, wie ich nun mal bin, habe ich mir auch die Google-Skill-Code-Dateien auf GitHub, die dieses Tutorial empfahl, angesehen. (<https://github.com/tartanguru/alexa-google-search>).

Ich eröffnete auch einen Developer-Account auf <https://developer.amazon.com/de/> für die Erstellung meines Skills und Sprachmodells und einen AWS-Account unter <https://aws.amazon.com/de/> für den eigentlichen Programmcode.

Eine mentale Hürde für viele ist das Hinterlegen der Kreditkartendaten für die Eröffnung des AWS-Accounts. Der YouTuber Venix versicherte mir, dass das hier eigentlich nur eine Formalie ist und man nichts für seine Skills bezahlen müsse.

Da mein Naturell relativ risikofreudig ist, habe ich seiner Aussage vertraut und es bis jetzt nicht bereut. Ich habe ohnehin nur eine Prepaid-Kreditkarte und konnte so einen eventuellen Schaden relativ sicher begrenzen.

Aber so viel vorweg, ich habe mithilfe des Internets bis heute keinen einzigen Cent an AWS bezahlt. Denn Amazon hat ein AWS-Promotional-Credits-Programm. Jeder Entwickler, der dort angemeldet ist, bekommt monatlich AWS-Credits im Wert von 100 Dollar geschenkt. Dieser Betrag reicht locker aus, um auch populäre Skills zu betreiben, und bei Bedarf werden diese Credits auch erneuert. Amazon will die Skill-Entwicklung und den -Betrieb für seine Entwickler kostenneutral halten.

Wer also Alexa Skills entwickeln möchte, sollte sich unbedingt hier anmelden:

<https://developer.amazon.com/de/alexa-skills-kit/alexa-aws-credits>.

Der Einstieg in die Alexa-Skill-Programmierung

In diesem Kapitel werden Sie einen Hosted-Skill direkt in der Alexa Developer Console einrichten. Dafür nutzen wir auch gleich schon mal das von Amazon vorgegebene »Fact Skill«-Template, das Sie nach Ihrem Geschmack anpassen können. Hier im Buch geht es um einen einfachen Fakten-Skill zum Thema Weltraum. Sie können natürlich aber auch ein ganz eigenes Faktenthema wählen. Das Prinzip bleibt das gleiche.

Um Ihren eigenen Skill zu programmieren, benötigen Sie als Erstes einen Alexa-Developer-Account, damit Sie loslegen können. Dafür gehen Sie auf die Seite <https://developer.amazon.com/alex/console/ask>.

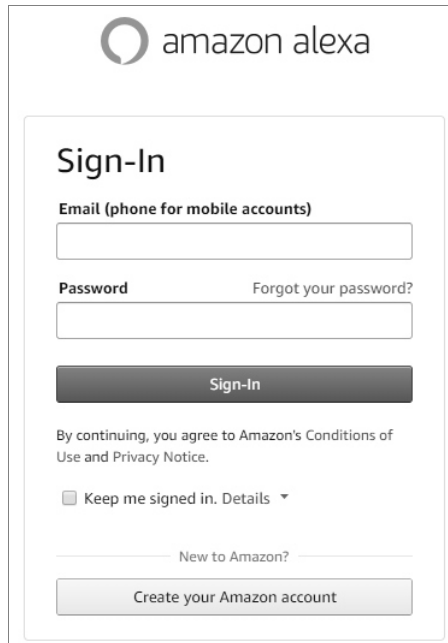
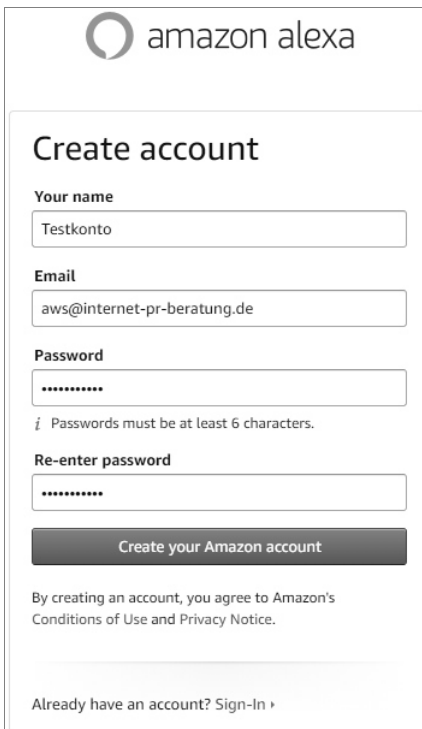


Abb. 1.1: Alexa-Developer-Console-Anmeldemaske

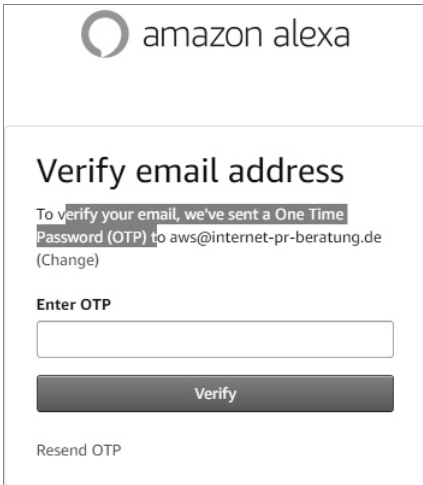
Klicken Sie nun auf **CREATE YOUR AMAZON ACCOUNT**.



The screenshot shows the 'Create account' page for Amazon Alexa. At the top is the Amazon Alexa logo. Below it, the title 'Create account' is displayed. The form contains four input fields: 'Your name' with the text 'Testkonto', 'Email' with 'aws@internet-pr-beratung.de', 'Password' with masked characters, and 'Re-enter password' also with masked characters. A small information icon and text state: 'i Passwords must be at least 6 characters.' Below the password fields is a dark blue button labeled 'Create your Amazon account'. Underneath the button, a line of text reads: 'By creating an account, you agree to Amazon's Conditions of Use and Privacy Notice.' At the bottom of the form, there is a link: 'Already have an account? Sign-In »'.

Abb. 1.2: Alexa Developer Console, CREATE ACCOUNT

Jetzt geben Sie Ihren Entwicklernamen sowie Ihre E-Mail-Adresse ein und vergeben ein Passwort mit mindestens 6 Zeichen. Danach klicken Sie unten auf den blauen Button.



amazon alexa

Verify email address

To verify your email, we've sent a One Time Password (OTP) to aws@internet-pr-beratung.de (Change)

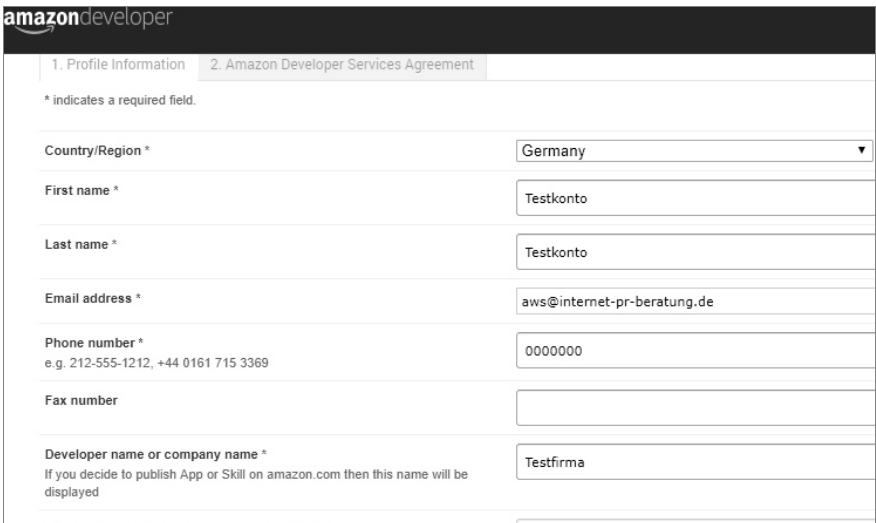
Enter OTP

Verify

Resend OTP

Abb. 1.3: E-Mail-Adresse bestätigen

Zu guter Letzt müssen Sie die Einrichtung mit einem *One-Time Password* (OTP) bestätigen.



amazon developer

1. Profile Information 2. Amazon Developer Services Agreement

* indicates a required field.

Country/Region * Germany ▼

First name * Testkonto

Last name * Testkonto

Email address * aws@internet-pr-beratung.de

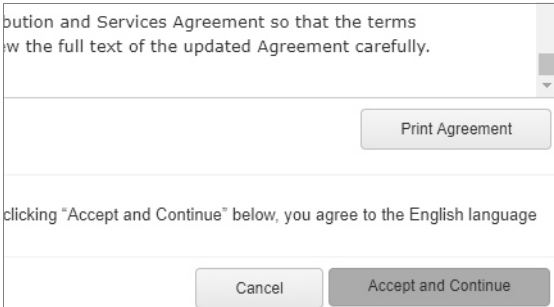
Phone number * 0000000
e.g. 212-555-1212, +44 0161 715 3369

Fax number

Developer name or company name * Testfirma
If you decide to publish App or Skill on amazon.com then this name will be displayed

Abb. 1.4: Developer-Registrierung

Jetzt müssen alle Pflichtangaben in dem Formular ausgefüllt werden. Sie legen hier auch den Firmennamen fest, falls Sie für eine Firma Skills entwickeln möchten. Wenn Sie alles gewissenhaft ausgefüllt haben, klicken Sie auf **ACCEPT AND CONTINUE**.



portion and Services Agreement so that the terms
w the full text of the updated Agreement carefully.

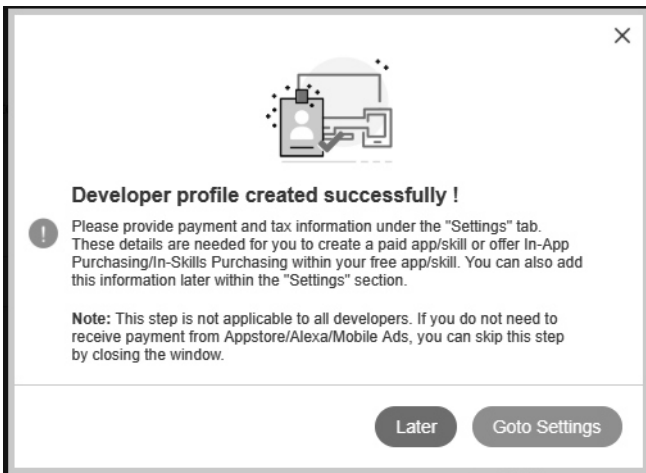
Print Agreement

clicking "Accept and Continue" below, you agree to the English language

Cancel Accept and Continue

Abb. 1.5: Amazon Developer Services Agreement

In dieser Maske klicken Sie für den Ausdruck des Vertrages auf **PRINT AGREEMENT** und anschließend auf **ACCEPT AND CONTINUE**.



Developer profile created successfully !

! Please provide payment and tax information under the "Settings" tab. These details are needed for you to create a paid app/skill or offer In-App Purchasing/In-Skills Purchasing within your free app/skill. You can also add this information later within the "Settings" section.

Note: This step is not applicable to all developers. If you do not need to receive payment from Appstore/Alexa/Mobile Ads, you can skip this step by closing the window.

Later Goto Settings

Abb. 1.6: Amazon Developer Console

Jetzt sind Sie in der Amazon Developer Console registriert. Sie haben nun die Möglichkeit, hier Ihre Steuerdaten und Kontoverbindungen für App- oder Skill-Einnahmen zu hinterlegen. Dann haben Sie einen vollwertigen Amazon-Developer-Account und können neben Skills auch Apps für Smartphones entwickeln, aber das ist ein anderes Thema.

Fahren Sie in der Amazon Developer Console oben in der Navigation auf ALEXA und klicken Sie anschließend im Untermenü auf ALEXA SKILLS KIT.

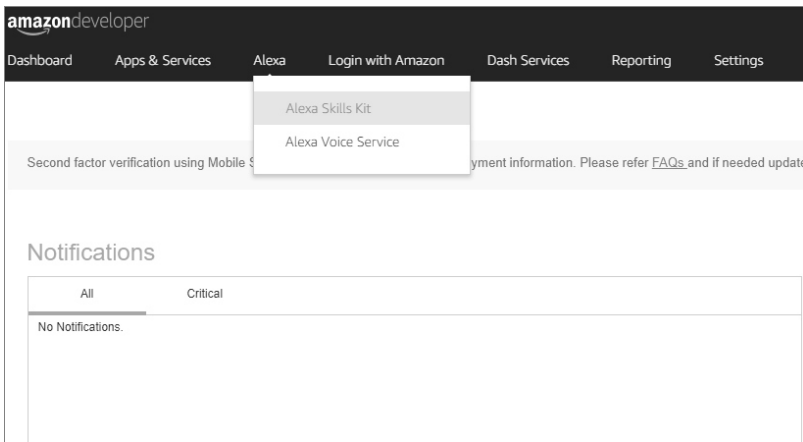


Abb. 1.7: ALEXA SKILLS KIT in der Navigation oben

Jetzt sind Sie in der Alexa Developer Console. Von hier aus überblicken Sie alle Ihre Skills, eventuelle Verdienste, Zahlungen sowie das Hosting, falls Sie »Alexa-Hosted« für Ihren Skill ausgewählt haben.

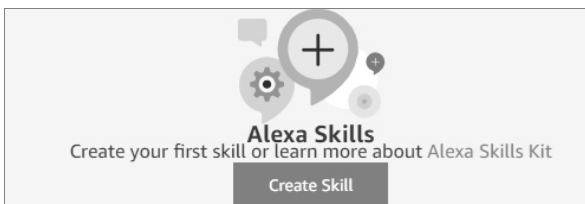


Abb. 1.8: Die Alexa Developer Console

Die Hürde, einen Skill zu entwickeln, hat Amazon immer weiter gesenkt und vereinfacht, indem Sie nun bei der Skill-Erstellung die Wahl haben: Sie können Ihren Programmcode in einer AWS Cloud, in einer privaten Cloud oder als »Alexa-Hosted«-Skill anlegen. Für den Einstieg wählen Sie die einfachste Variante, einen **Alexa-Hosted-Skill**.

Create a new skill

Skill name
Testskill
9/50 characters

Default language
German (DE) ▼
More languages can be added to your skill after creation

Choose a model to add to your skill
There are many ways to start building a skill. You can design your own custom model or start with a pre-built model. Pre-built models are interaction models that contain a package of intents and utterances that you can add to your skill.

Custom	Flash Briefing	Smart Home	Video
Design a unique experience for your users. A custom model enables you to create all of your skill's interactions.	Give users control of their news feed. This pre-built model lets users control what updates they listen to. "Alexa, was ist in den Nachrichten?"	Give users control of their smart home devices. This pre-built model lets users turn off the lights and other devices without getting up. "Alexa, schalte das Licht ein"	Let users find and consume video content. This pre-built model supports content searches and content suggestions. "Alexa, suche nach Komödien"

Choose a method to host your skill's backend resources
You can provision your own backend resources or you can have Alexa host them for you. If you decide to have Alexa host your skill, you'll get access to our code editor, which will allow you to deploy code directly to AWS Lambda from the developer console.

Provision your own	Alexa-Hosted
Provision your own endpoint and backend resources for a skill. With this option, you will not gain access to the console's code editor.	Alexa will host skills in your account up to the AWS Free Tier limits. You will gain access to an AWS Lambda endpoint, 5 GB of media storage with 15 GB of monthly data transfer, and a table for session persistence. Learn more

Abb. 1.9: Screenshot aus der Alexa Developer Console

Der Vorteil der Alexa-Hosted-Skills ist, dass Sie sich als Entwickler keine Sorgen darum machen müssen, wie die Cloud betrieben wird. Amazon stellt den Nutzern eine fertige Umgebung für den Skill zusammen, der dem Betreiber nicht berechnet wird. Sie müssen hier auch keine Kreditkartendaten hinterlegen. Ich werde in Kapitel 6 näher

darauf eingehen, wie Sie einen Skill selbst über die AWS Cloud einrichten und betreiben. Aber fangen wir zunächst mit den Grundlagen an.

Wenn Sie keine Scheu vor Programmcode haben, neue Dinge lernen möchten oder gar schon selbst etwas in JavaScript programmiert haben, ist die Seite <https://github.com/alexa/> der richtige Einstiegspunkt für Sie.

Dort finden Sie fertige Alexa Skills als Vorlagen für Ihren eigenen Skill. Auch das Hosted-Skill-Codebeispiel befindet sich dort. Im Prinzip brauchen Sie hier nur noch im Code die Texte für die Sprachausgabe und Textanzeige zu verändern. Das einfachste Beispiel für den Einstieg ist wohl das Code-Beispiel für einen »Fact Skill« unter <https://github.com/alexa/skill-sample-nodejs-fact>. Ich habe so angefangen, indem ich mir die `index.js`-Datei des Code-Beispiels im Browser angeschaut und versucht habe, die Stellen zu finden, wo ein Text ausgegeben wurde. Anschließend habe ich ihn an meine Idee angepasst.

Zudem empfehle ich Ihnen, sich im Netz über JavaScript fortzubilden. Über eine Google-Suche nach »JavaScript Tutorial« werden Ihnen viele nützliche Quellen angezeigt, um die Grundlagen von JavaScript zu lernen.

1.1 Was ist ein Alexa Skill?

Ein Alexa Skill ist im Prinzip eine Voice-App, nur dass diese nicht auf einem Gerät installiert, sondern in der Cloud betrieben wird. Ein Skill wird also nicht installiert, sondern nur aktiviert.

Das hat für den Nutzer den Vorteil, dass er keine Ressourcen für die Speicherung einer Applikation auf seinem Gerät benötigt. Dieser Vorteil beinhaltet aber auch gleichzeitig den Nachteil, dass ein Skill nur mit einer Internetverbindung zur Cloud funktionieren kann.

Alexa wird mithilfe von Skills schlauer, aktuell gibt es international über 100.000 Skills. In Deutschland sind es gerade mal etwas über 11.600.

Die beliebtesten Kategorien sind in Deutschland Musik & Audio (2989), Spiele & Quiz (1604), Bildung & Nachschlagewerke (945), Lifestyle (891) und News (791), Stand 11.11.2019. Wie eine Case Study der Website [beyto.com](https://www.beyto.com) schreibt, ist das Skill-Wachstum in Deutschland eher gebremst, siehe <https://www.beyto.com/wachstum-skills-amazon-alexa-skill-store-oktober-2019/>.

1.1.1 Wie wird ein Skill aufgerufen?

Per Sprache wird dem Amazon-Echo-Gerät eine Frage oder Aufforderung mitgeteilt. Ein Nutzer könnte zum Beispiel sagen: »Alexa, öffne *Bierflasche*.« Dieser einfache Satz wird von der KI in die Bestandteile zerlegt und interpretiert:

[Wake Word], [Launch Word] [Invocation Name] [Connecting Word] [Utterance]

Diese Wake Words kennt Amazon Echo:

- Alexa
- Amazon
- Computer
- Echo

Das Wake Word (Weckwort) muss vom Nutzer über die Alexa-Companion-App eingestellt werden. Standardmäßig ist »Alexa« voreingestellt. Bisher ist Alexa der einzige Heimassistent, der den Nutzern hier eine Wahlmöglichkeit lässt, das ist einer der Gründe, warum Alexa so beliebt ist.

Beispiel

Alexa, öffne Bierflasche und gib mir einen Trinkspruch.

Diese »Launch«-Wörter kennt Amazon Echo:

- Start
- Starte
- Öffne
- Frage
- Lade
- sprich mit
- führe (Ausführungsname) aus
- benutze

Diese Kommandos sollten Sie im Hinterkopf behalten, wenn Sie einen »Invocation Name« für Ihren Skill auswählen, um eine möglichst natürliche Kommunikation zu ermöglichen.

Beispiel

Alexa, **öffne** Bierflasche und gib mir einen Trinkspruch.

Was ist ein Invocation-Name?

Der Invocation-Name bzw. Ausführungsname ist der Name des Skills. Der Skill-Name ist fast frei wählbar. Er sollte zu einer natürlichen Sprache passen. In unserem Beispiel lautet der Ausführungsname **Bierflasche**.

Wenn also der Bierflasche-Skill aktiviert wurde, erkennt Alexa ihn und kann zu der Aufforderung »Alexa, öffne Bierflasche« eine sinnvolle Antwort geben.

Beispiel

Alexa, öffne **Bierflasche** und gib mir einen Trinkspruch.

»Connecting«-Wörter für Skills:

- mit
- und
- aus
- um
- für

Beispiel

Alexa, öffne Bierflasche **und** gib mir einen Trinkspruch.

Was bedeutet »Utterance«?

»Utterance« ist das englische Wort für Äußerung. Äußerungen können bestimmte Intents zugeordnet werden.

Beispiel

Alexa, öffne Bierflasche und **gib mir einen Trinkspruch**.

Was bedeutet Intent?

Wenn wir miteinander kommunizieren, haben wir immer eine Intention (Absicht), warum wir kommunizieren. Wenn wir etwas Bestimmtes wissen wollen, fragen wir also unseren Gesprächspartner etwas. In der Skill-Programmierung wird der Begriff *Intent* genauso verwendet. Mit einem Alexa-Skill-Intent soll also eine bestimmte Absicht des Anwenders befriedigt werden.

Wenn Sie einen Skill programmieren möchten, sollten Sie sorgfältig alle Nutzungsszenarios Ihres Skills planen und jedem Szenario einen Intent zuordnen. In dem vorangegangenen Beispiel meines Bierflasche-Skills wurde die Äußerung »gib mir einen Trinkspruch« dem Intent »Trinkspruch« zugeordnet, der genau diesen Wunsch des Nutzers erfüllt. Die Äußerung »gib mir ein Bier« würde das Geräusch

einer aufploppenden Bierflasche abspielen und ist demzufolge dem Bierflaschen-Intent zugeordnet.

1.1.2 Wie funktioniert ein Skill genau?

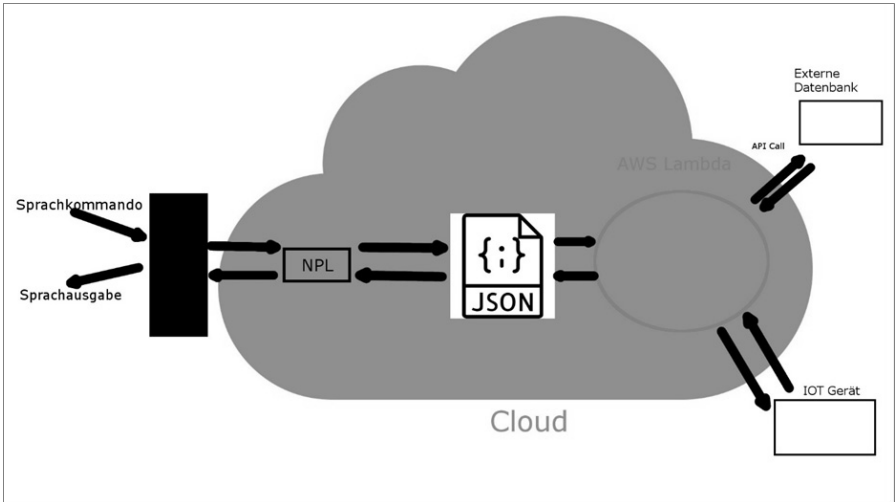


Abb. 1.10: Funktionsschema eines Alexa Skills

Das Schema in Abbildung 1.10 zeigt den theoretischen Aufbau eines Alexa Skills. In dem Beispiel gibt ein Nutzer ein Sprachkommando an Alexa. Dieses Sprachkommando wird durch NLP, Natural Language Processing, als ein bestimmter »Intent« erkannt, der im Sprachmodell zum Skill hinterlegt ist und in maschinenlesbaren JSON-Code umgewandelt wird. Ein Intent ist hier also eine Nutzerabsicht, die im Sprachmodell hinterlegt ist und durch bestimmte Äußerungen des Nutzers eine bestimmte Reaktion des Skills auslöst. Der JSON-Code wird zur Ausführung an den Programmcode in der AWS Lambda zur Verarbeitung gesendet.

Je nach Skill und Programmcode wird entweder eine externe Datenbank als Antwortquelle abgefragt, ein IoT(Internet of Things)-Gerät angesprochen oder eine im Code hinterlegte Antwort/ein Ergebnis berechnet. Diese wird dann im JSON-Format zurückgesendet und

durch die NLP Unit in für Menschen verständlicher Sprache ausgegeben.

Als Skill-Programmierer haben Sie Einfluss auf das Sprachmodell des Skills. Dieses kann direkt in der Alexa Skill Console erstellt werden.

Die zweite wichtige Komponente ist die Skill-Logik, diese wird als Code, idealerweise als AWS-Lambda-Funktion, ausgeführt. Sie müssen aber Ihren Skill-Code nicht unbedingt in der AWS Cloud hosten. Ein per HTTPS verschlüsselter Server genügt auch. Der Einfachheit halber beschränke ich mich in diesem Buch auf Beispiele, die primär die AWS-Clouddienste nutzen.

1.2 Den Skill in der Alexa Developer Console anlegen

Die Sprache festlegen

Um einen neuen Skill in der Alexa Developer Console anzulegen, klicken Sie oben auf BUILD. Zuerst legen Sie die Sprache für Ihren Skill fest.

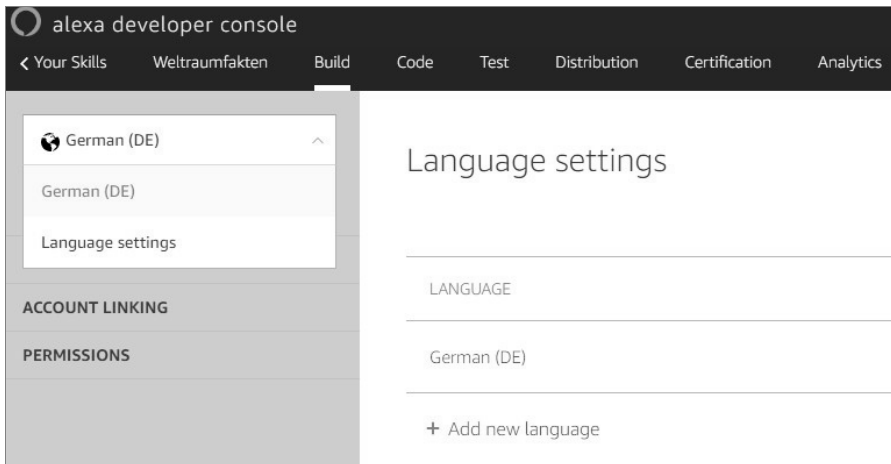


Abb. 1.11: Sprachmodell auswählen und festlegen

Den Invocation-Namen erstellen

Danach erstellen Sie den *Invocation*-Namen, der den Skill starten soll.

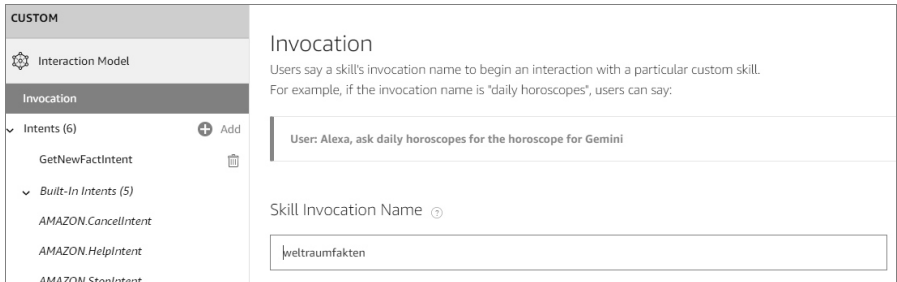


Abb. 1.12: INVOCATION NAME erstellen

Einen eigenen Intent erstellen

Anschließend können Sie mit dem Erstellen eigener Intents oder dem Einbinden von vorinstallierten Amazon-Intents fortfahren.

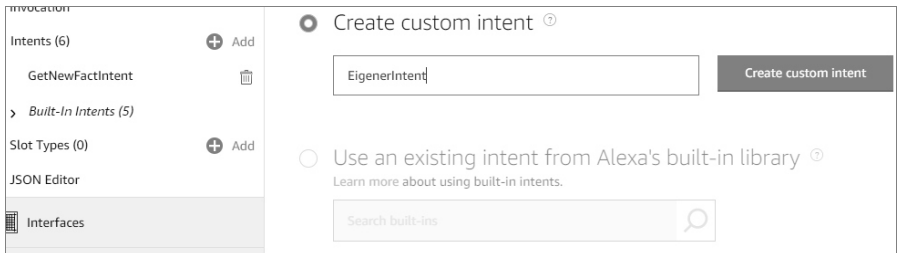


Abb. 1.13: Intent erstellen

Eigene Trainingsäußerungen erstellen

Nun können Sie für Ihren »EigenerIntent« Ihre Äußerungen hinterlegen. In diesem Beispiel habe ich mich dafür entschieden, auch Slots zu verwenden.

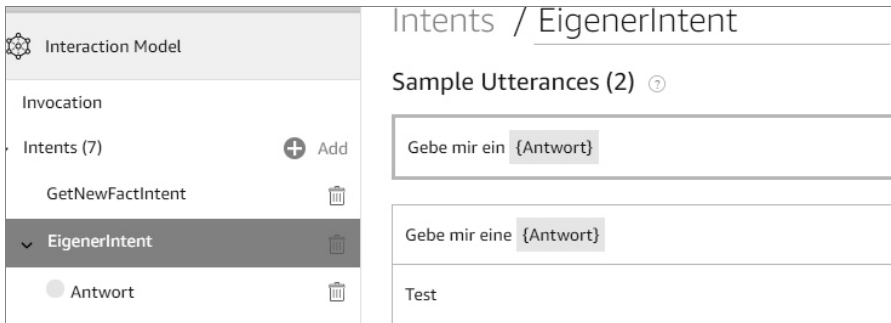


Abb. 1.14: Trainingsäußerungen erstellen

Wenn Sie alle Äußerungen für diesen Intent eingegeben haben, klicken Sie oben zunächst auf **SAVE MODEL** und dann auf **BUILD MODEL**.

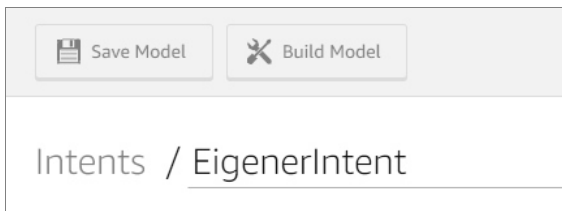


Abb. 1.15: SAVE MODEL und BUILD MODEL

1.2.1 Slot-Typ erstellen

Bei meinen Äußerungen hier im Beispiel habe ich mit `{Antwort}` einen Slot mit dem Namen »Antwort« definiert (siehe Abbildung 1.14). Slots sind wie Variablen, die in einer Äußerung ausgetauscht werden können. Durch die Nutzung von Slots in den Äußerungen können Sie eine Vielzahl von Äußerungen mit geringem Aufwand erstellen.

Diesen Slots wiederum werden Slot-Typen zugeordnet. Im Folgenden zeige ich Ihnen, wie das geht.

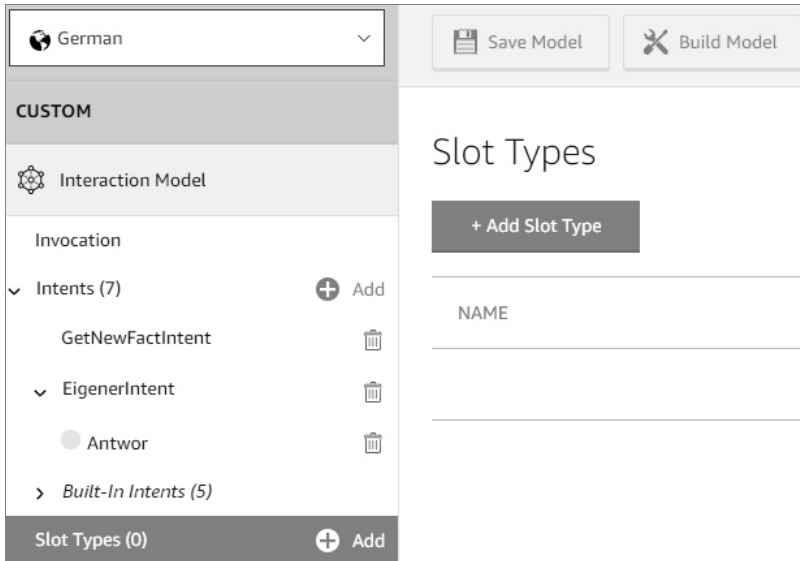


Abb. 1.16: Slot-Typ erstellen, Schritt 1

Nachdem Sie auf ADD SLOT TYPE geklickt haben, können Sie auswählen, ob Sie einen eigenen neuen Slot Type erstellen oder einen vortrainierten nutzen möchten.

Amazon bietet mittlerweile über 49 vortrainierte Slot Types an. Das geht von Slot Types für Schauspieler, Fluglinien, Flughäfen, Tiere, Künstler, Bücher bis hin zu Regionen, Städte, Länder und viele mehr. In diesen vortrainierten Slot-Typen sind die gängigsten Begriffe zum Slot-Typ schon voreingetragen. Sie können diese aber meistens weiter ergänzen, falls mal ein Begriff nicht eingetragen ist.

☒ Create custom slot type [?]

Create custom slot type

☐ Use an existing slot type from Alexa's built-in library [?]

Learn more about using built-in slot types.

Name	
> List Types 43 built-ins	These slot types each
> Numbers, Dates, and Times 6 built-ins	These slot types tha

Abb. 1.17: Slot-Typ erstellen, Schritt 2

Slot Types / ANTWORTEN

Slot Values (3) [?]

Bulk Edit Export Search

VALUE [?]	ID (OPTIONAL) [?]	SYNONYMS (OPTIONAL) [?]	
Spruch	Enter ID	Add synonym	+
Aussage	Enter ID	Add synonym	+ Testaussage x
Antwort	Enter ID	Add synonym	+ Testantwort x

Abb. 1.18: Slot-Typ erstellen, Schritt 3

Wenn Amazon keinen relevanten Slot Type für Ihren Skill vorrätig hat, können Sie ganz einfach einen eigenen Slot Type definieren, indem Sie den Namen des neuen Slot-Typs in das Eingabefeld unter **CREATE CUSTOM SLOT TYPE** eintragen und anschließend auf den gleichnamigen Button daneben klicken (Abbildung 1.17).

In meinem Beispiel habe ich die Slot-Typen »Antworten« mit den Werten »Spruch«, »Aussage« und »Antwort« erstellt.

Nun versteht mein Skill also die Äußerungen:

- »gebe mir eine Antwort«,
- »gebe mir eine Aussage« und
- »gebe mir einen Spruch«.

Sie können hier auch Synonyme für bestimmte Begriffe hinterlegen. Dies verbessert nochmals die Erkennungsrate.

Zudem können Synonyme und IDs auch nützlich sein, um bestimmte Parameter zu erkennen, die normalerweise kein Mensch sagen würde. Diese Parameter können dann für bestimmte Variablen im Programmiercode eingesetzt werden, um beispielsweise eine bestimmte API-URL abrufen zu können.

Beispiel

Eine Wetter-API erfordert die Postleitzahl für die Abfrage der Wetterdaten. Anstatt den Nutzer zu nötigen, eine Postleitzahl zu sagen, kann er ganz natürlich die Stadt nennen, denn im Sprachmodell wurde im Feld ID die Postleitzahl hinterlegt. Der Code übergibt für die Wetterdaten-Anfrage die ID mit der Postleitzahl anstatt des Städtenamens, mit dem die Wetter-API nichts anfangen könnte.

1.2.2 Das JSON-Sprachmodell

Sie können Ihr JSON-Sprachmodell für den Skill ganz komfortabel in der Alexa Developer Console automatisch erstellen lassen.

Die Alexa Developer Console erstellt dieses Modell, nachdem Sie all Ihre Intents, Trainingsäußerungen und eventuellen Slots mit Slot-Typen angelegt haben, wenn Sie auf SAVE MODEL und anschließend auf BUILD MODEL geklickt haben.

Sie können aber auch, anstatt zuerst alles manuell in der Alexa Developer Console zu erstellen, die JSON-Sprachmodelle der Codebeispiele von GitHub direkt in den JSON-Editor der Alexa Developer Console kopieren und das Model nachher Ihren Anforderungen anpassen. Das spart Zeit!

Bei jedem Skill-Beispiel auf GitHub ist auch das Sprachmodell mit dabei, für unser Beispiel ist es hier zu finden: <https://github.com/alexa/skill-sample-nodejs-fact/blob/master/models/de-DE.json>. Sie sollten sich das Sprachmodell zu Beginn intensiv anschauen, um zu verstehen, wie ein Nutzer mit dem Beispiel-Skill interagieren kann.

Amazon gibt dem Entwickler Pflicht-Intents vor, die von Amazon selbst vortrainiert sind. Zu Pflicht-Intents gehören **AMAZON.CancelIntent**, **AMAZON.HelpIntent** und **AMAZON.StopIntent**. Zudem gibt es noch optionale Intents von Amazon, die auch vortrainiert sind. Als Entwickler können Sie diese meistens mit weiteren Äußerungen ergänzen. Ich schreibe deswegen »meistens«, weil das bei manchen vortrainierten Intents nicht möglich ist. So ist eine weitere Ergänzung des Intents Amazon.Number, sinnlos und nicht möglich, da für diesen Intent schon alle Zahlen, die es gibt, vortrainiert sind.

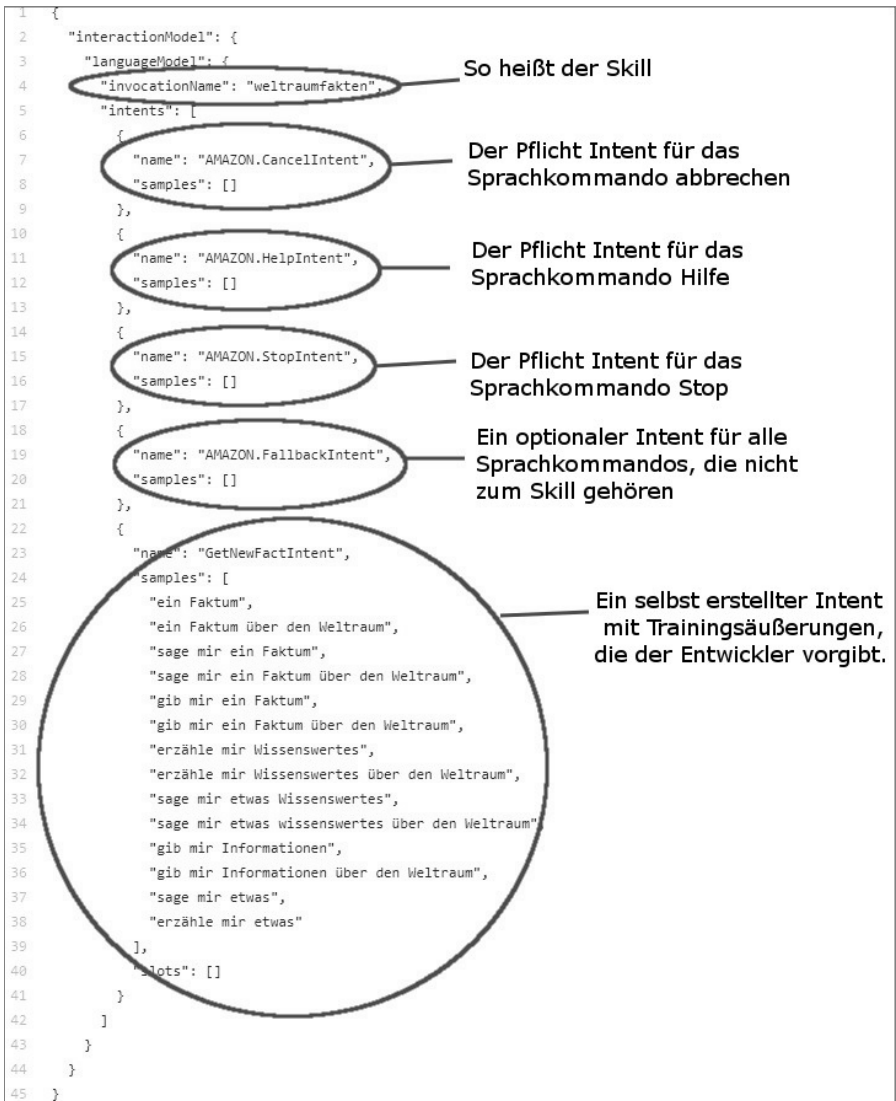


Abb. 1.19: Ein Alexa-Sprachmodell in JSON-Format

Stichwortverzeichnis

A

- Abfrageprozess 135
- Abonnement 311
- Abwechslungsreiche Antworten 307
- Accept only a set of values 139
- Accept only Slot Type's values and synonyms 140
- Account-Linking 82, 294
- AGB 196
- Alexa App 99
 - Antwortkarten 43
- Alexa Developer Console 23, 111
- Alexa for Business Organizations 259
- Alexa Presentation Language (APL)
 - 43, 112, 126
- Alexa Skill 29
- Alexa Skill Store 255
- Alexa Skills Kit 47, 87, 172
- Alexa Skills Kit Command Line Interface 153
- Alexa speech prompts 136
- Alexa-Companion-App 30, 196
- Alexa-Developer-Account 23
- Alexa-Handler 47
- Alexa-Hosted 28
- Alexa-Hosted-Skill 153, 198
 - Debugging 82
- Alexa-Live-Event 141
- Alexa-Newsletter 278
- Alexa-Simulator 58
- Alexa-Skill-Marketing-Guidelines 290
- Alltägliche Situation 305
- Alphabet 92
- Amazon CloudWatch Logs 164
- Amazon Customer Profil API 232
- Amazon Developer Console 27
- Amazon Pay 297, 311
- Amazon Ressource Number 164
- Amazon Rewards 298
 - Kategorien 298
- Amazon S3 Bucket 119
- Amazon Shop 294
- AMAZON.CancelIntent 66, 122
- AMAZON.HelpIntent 66
- AMAZON.LoopOffIntent 122
- AMAZON.LoopOnIntent 122
- AMAZON.MoreIntent 122
- AMAZON.NavigateSettingsIntent 122
- AMAZON.NextIntent 122
- AMAZON.NoIntent 66
- AMAZON.PageDownIntent 123
- AMAZON.PageUpIntent 122
- AMAZON.PreviousIntent 122, 123
- AMAZON.RepeatIntent 65, 122
- AMAZON.ScrollDownIntent 123
- AMAZON.ScrollLeftIntent 123
- AMAZON.ScrollRightIntent 123
- AMAZON.ScrollUpIntent 123
- AMAZON.ShuffleOffIntent 122
- AMAZON.ShuffleOnIntent 122
- AMAZON.StartOverIntent 122
- AMAZON.StopIntent 66
- AMAZON.YesIntent 66
- Amazon-Marketing-Programm 278
- Amazon-SEO-Faktor 277
- Amazon-Zertifizierungsteam 195
- Analyse
 - semantische 276
- Analytics 302
- Anfrage-Handler 52

- Anfrage-Interceptor 52
 - Antworten
 - abwechslungsreiche 307
 - Antwortkarte 43, 113, 212
 - AskPermissionConsent 114
 - LinkAccount 113
 - Simple 113
 - Simple Card 61
 - Standard 113
 - Standard Card 61
 - URLs 121
 - Antwortquelle
 - externe Datenbank 33
 - Antwortschema 129
 - API 195, 210
 - API Request 232
 - API-Bibliothek 218
 - API-Endpoint 233
 - API-Key 218
 - API-Post-Request 231
 - API-Schlüssel 219
 - API-URL 39
 - APL 43
 - Application Programming Interface 210
 - ARN *siehe* Amazon Ressource Nummer
 - Array 48
 - ASK CLI
 - Befehle 182
 - installieren 181
 - ASK-CLI 153, 172
 - ASK-SDK 52, 113, 145, 198
 - ASK-SDK 2 76
 - ASK-SDK-Core 46
 - ASK-Toolkit
 - installieren 184
 - Ordnerstruktur 186
 - async 200
 - Asynchron 221
 - Asynchrone Operation 222
 - Attribut
 - persistentes 65, 75
 - Sitzungs- 65
 - AttributesManager 77
 - Audacity 310
 - Audio Player 112
 - Audiodatei 75, 310
 - Audiofile 89
 - Restriktionen 89
 - Audio-Player 121
 - Aufmerksamkeitsmechanismus 272
 - Ausführungsname 31
 - Äußerung
 - emotionale 307
 - Äußerungsrepräsentationsvektor 273
 - Auto Delegation 43, 112, 126, 129
 - Automatische Delegation 134
 - await 222
 - AWS 19
 - Account 19
 - Credits 19
 - AWS Cloud 29, 34
 - AWS CloudWatch 82, 159
 - AWS Console 155
 - AWS Credit 111
 - AWS Dynamo DB 159
 - AWS IAM 180
 - AWS Lambda 33, 159
 - Code-Editor 164
 - einrichten 159
 - Vorlage 162
 - AWS Managementkonsole 83
 - AWS Polly 97
 - AWS Promtional Credits 101
 - AWS S3 159
 - AWS S3 Bucket 101
 - AWS Service Amazon Polly 310
 - AWS-Account 154
 - AWS-Cloud-Infrastruktur 153
 - AWS-IAM-Benutzer 174
 - AWS-Konto
 - erstellen 155
 - AWS-Profil 182
 - AWS-Promotional-Credits 19
- B**
- Backend-Logik 174
 - Beispielaussage 283

Berechtigungen 178
 Berechtigungsantwortkarte 201
 Berechtigungswert 209
 Beta-Tester 259
 Beta-Test-Phase 259, 260
 Betonung 90
 Bierflasche-Skill 31
 Bildformat 118
 Bi-LSTM-Schicht 276
 Blueprint 162
 Browsererweiterung 220
 BUILD MODEL 40
 Build-In Intent 65

C

Callback-Methode 222
 CanFulfillIntentRequest 270
 Card Title 212
 Case 265
 catch 223
 Cloud 29
 CMS 97
 Code Snippet
 generieren 186
 Codebeispiel 29
 Code-Editor 153
 Command Line Interface 172
 Command-Prompt 170
 COMPLETED 132
 confirmIntent 132
 confirmSlot 132
 consentToken 85, 200
 Consumable 312, 313, 320
 Code 323
 speichern 322
 Content Guidelines 265
 Content-Backend 117, 210
 als Tabelle 211
 Context Switching 134
 CORS 106, 119
 CORS-Konfiguration 106
 Crosshypothese-Feature 276
 Cross-Origin Resource Sharing 106,
 119

currentIntent 146
 Custom Interface 125
 Custom Interface Controller 112
 CUSTOM SLOT TYPE 39
 Customer API 195
 Customer Profile API 200
 CVS-Datei
 Beta-Test 259

D

Daten
 abrufen 195
 persönliche 258
 Datenblöcke 225
 Datenschutzerklärung 196
 Debugging 82, 150
 Delegation
 automatische 134
 Detailed description 318
 Dialog 129
 Dialogfluss 146
 Dialogmanagement 129
 Dialogmanagement-Skill 295
 DialogManagementStateInterceptor
 151
 Dialogmodell 126
 Dialogscript 142
 dialogState 132
 Dialogstrategie 133
 disable auto delegation 134
 Display-Interface 112, 122
 Domäne 271
 Domänenklassifizierung 274
 DynamoDB 76, 327
 DynamoDB-Tabelle 79

E

Earnings 302
 Echo-Button 112, 124
 E-Commerce 293
 Einkaufservice per Sprache 293
 Einmalkauf 314
 elicitSlot 132
 E-Mail Intent 235

E-Mail-Adresse 195
 abfragen 196
 EmailIntentHandler 199
 kompletter Code 204
 E-Mail-Signatur 283
 E-Mail-Template
 definieren 238
 E-Mail-Versand 238, 249
 Emotionale Äußerung 307
 enable auto delegation 134
 Encoder-Netzwerk 272
 Endpoint 44, 111, 119, 164
 Endpoint-URL 219
 Engagement 302
 ErrorHandler 83
 Example Phrases 257
 Export Compliance 259
 Expressbestellung 295
 Externe Daten abrufen 195

F

Facebook 280, 284
 Gruppen 284
 Fact Skill 23
 Fallback to skill setting 134
 Functional Test 262, 263

G

Gadget Controller 124
 Game Engine 124
 Geolokalisierungsdienst 210
 Geschäftsmodell 278
 GitHub 40, 314
 Google 278
 Google API Console 217
 Google Sheet API 217
 Google Sheets 210
 Google Tabelle 211
 ID 216
 URL 214
 GPS 85
 GPS-Daten 195
 GUI 124
 Guidelines 263, 289
 Marketing 289

H

Handler 47
 Happy Path 142
 Hashtags 287
 Haupt-Code-Ausführungsthread 221
 Hosted-Skill 23
 HTTP-Modul 223
 HTTP-Requests 210
 HTTPS 100
 HTTPS-Abfrage 235
 Humor 266
 Hypothesengenerator 276
 HypRank 273

I

i18next-Bibliothek 48
 Icon-Grafik
 Skill-Store 257
 ID 39
 IDE (integrated development environment) 153, 184, 232
 if-Statement 150
 IN_PROGRESS 132
 Informationen
 sammeln 129
 Informationsverarbeitung 113
 In-Skill Purchase 14, 259
 In-Skill-Kaufart 321
 In-Skill-Produkt
 erstellen 317
 In-Skill-Purchasing (ISP) 311
 Integrated Development Environment 153
 Intent 32
 Sprachmodell hinzufügen 67
 Intent History 140
 Intent-Handler 150, 190
 IntentRequest 74
 Intent-Vorlage 188
 Interaktionsmodell
 Fehler 316
 Interaktionsrate 305
 Interface 43
 Audio Player 121
 Auto Delegation 126

- Custom Interface Controller 125
- Display 122
- Echo Button 124
- VideoApp 123
- Invocation 35
- Invocation Name 31, 278
- ISP *siehe* In-Skill Purchase

J

- JavaScript 29
 - Laufzeitumgebung 172
- JavaScript Tutorial 29
- JSON
 - Sprachmodell 40
- JSON API Endpoint URL 212
- JSON Editor 111
- JSON Rest-API 216
- JSON Viewer 220
- JSON-Antwort 48
- JSON-Code 33
- JSON-String 250

K

- Kategorie
 - Skill-Store 257
- Keyword
 - Skill-Store 257
- KI 30
- Kids-Skill 259
- Kind 259
- Klassifizierungsmodell 273
- Klassifizierungsnetzwerk 272
- Konstante 48
- Kontextinformation 277
- Kontextwechsel 130, 143
- Konversation 43
- Kreditkartennummer 234
- Künstliche Intelligenz 11
- Kurzbeschreibung 255

L

- Lambda-Funktion 52
- Langzeitgedächtnis 75
- Langzeitspeicherung 322

- largeImageUrl 118
- Laufzeitlatenz 273
- Launch-Phrase 257, 278
- LaunchRequest 47
- Lernen
 - maschinelles 270
- Let's-crypt-Zertifikat 119
- Listen-Ranking-Ansatz 276
- Logfile 82
- LogIncomingRequestInterceptor 85
- LSTM-Modell 276

M

- Machine-Learning-Modell 270
- macOS 182
- Mailgun 232
- mailgun-js-Modul 233
- Maschinelles Lernen 270
- Maßeinheit 195
- Mehrsprachigkeit 46
- Meta-Keyword-Tag 271
- Monetarisierungsstrategie 313
- Monetarisierungstool 317
- MP3-Datei 310
- MS Visual Studio Code 153, 184, 232

N

- Natural Language Processing 33
- Netzwerk 273, 283
- Newsletter 283
- NLP 33
- Node Version Manager 172
- Node.js
 - installieren 166
 - LTS 166
 - SSML-Tags 98
- Node.js-Modul 235
- NoSQL 76
- Not within a set of values 141
- Notepad++ 153
- npm 181
- npm-Paket-Manager 167
- Nutzeräußerung 271
- Nutzerbewertung 277
- Nutzerinformationen 195

Nutzerkreis
 einschränken 259
Nutzungsstatistik 302
Nutzungsszenario 32

O

Objekt-URL 111
One sentence description 318
One-Shots 257
One-Shot-Utterance 131
One-Time Password 25, 301
One-Time Purchase (OTP) 312
Onlineshop 294
Operation
 asynchrone 222
Ordner 233
OTP 301

P

package.json 199
Parameter 39
permission values 209
PersistenceAdapter 76
Persistentes Attribut 65, 75
Persistenzadapter 322
Personalisieren 195
Persönliche Daten 258
Pflicht-Intent 40
Phonem 92
Polly-Stimme 311
Post-Request 250
PowerShell 181
Preis
 festlegen 314
Premium-Content 14, 307
Premium-Fakten
 erstellen 314
Prepaid-Kreditkarte 154
Privacy Policy URL 257
Privatadresse 195
Produkt
 kaufen 293
Produktionssystem 273
Programmierfehler 262
Promise 222

Prüfergebnis 238
Prüfregel 138
Prüfungsinstanz
 Skill-Vorschläge 271
Purchase confirmation description
 319
Purchase prompt description 319

Q

Query 224

R

Ranking 255
Referenzname 317
reject 222
Reject only a set of values 139
Request Interceptor 48
Request-Handler 150
 registrieren als 208
RequestInterceptor 150
Re-Ranking 274
resolved 222
Response-Interceptor 150
Ressource 29
Revisionerzeugung 186
Richtlinie
 erstellen 175
Rüstzeug 231

S

S3 Bucket 101
SAVE MODEL 40
Schnittstelle
 benutzerdefinierte 125
SDK 98
Semantische Analyse 276
Semantische Interpretation von
 Äußerungen 275
SEO 269
sessionAttributes 146
SessionEndedRequest 84
SessionEndedRequestHandler 84
Settings API 195
Shopsystem 294
Shortlister 272

Shortlisting-Re-Ranking-Ansatz 272
 Simple Card 61, 116
 Situation
 alltägliche 305
 Sitzungsattribute 65, 145
 Skill
 hochladen 190
 personalisieren 75
 testen 183
 theoretischer Aufbau 33
 veröffentlichen 255
 Skill Card 100, 113
 Bildformat 118
 skill.json 190
 Skill-Beschreibung 277
 SkillBuilder 76, 208, 246
 Skill-Code-Datei 189
 Skill-Einnahme 301
 Skill-Gedächtnis 159
 Skill-Hypothese 276
 Skill-Icon 282
 Best Practices 282
 Skill-Katalog 255
 Skill-Logik 34
 Skill-Management-API 173
 Skill-Name 31, 270
 Skill-Promotion 269
 Skill-Szenario 235
 SkillTable 322
 Skill-User-ID 82
 Skill-Vergütung 305
 Skill-Vorschlag 269
 Skill-Wachstum in Deutschland 30
 Slot 35, 129
 Prüfung 140
 Slot-Typ 37
 Slot-Wert 131
 smallImageUrl 118
 Smart-Display 293
 Smartphone 99
 Soundbibliothek (Amazon) 310
 Soundfile 89, 309
 Spammer 234

Speech Synthesis Markup Language
 87
 Speechcon 212, 307
 Sprachbenutzeroberfläche 316
 Sprachbibliothek 49
 Sprachcomputer 130
 Sprachinteraktion 129
 Sprachkommando 33
 Sprachmodell
 Intent hinzufügen 67
 Sprachsuche 278
 SSL 111
 SSML 87
 SSML-Tag 58, 87, 212
 Antwortkarte 120
 STARTED 132
 Steuerinformationen 300
 Subdomain 111
 Suchmaschine 278
 Synonym 39

T

Tabelle
 veröffentlichen 213
 Tausender-Kontakt-Preis 305
 Tax Identity 300
 Telefonnummer 195
 Temperatureinheit 195
 Template 23
 Terms of User URL 257
 Test
 automatisiert 262
 Textantwort
 Tipps 120
 then-Methode 223
 TKP 305
 Trigger 164
 try/catch 240
 Twitter 287
 Twitter-Feed 210

U

Umschulungszyklus 273
 Unterordner 233

Upsell 312
URL
 in Antwortkarten 121
UserId 85
UserId-Wert 327
Utterance 32, 131, 305

V

Validation 138, 261
Validation-Regel 140
Vektor 273
Verbrauchsgut
 digitales 320
Vergütung 302, 305
Verzeichnisbaum 233
Video App 112
VideoApp-Interface 123
Videoformat 123
Videoinhalt 123
 Auflösung 124
Voice Commerce 100, 307
Voice Search 12
Voice User Interface 316
Voice-App 29
VS Code 184
 Dokumentation 185
VUI 316

W

W3C 87
W3C-Konsortium 106
Wake Word 30
Webprogrammierung 222
Website 284
Website-Ranking 277
Webspeicher 100
Weltraumfakten 42
Werbebudget 278
Werbeeinnahme 278
Wetterdienst 210
Wissensdatenbank 221
WordPress 97

X

XML 87

Y

YouTube 288

Z

Zeilenumbruch 121
Zeitzone 195
Zertifizierung 259, 261
 Antwortkarten 120
Zertifizierungsprozess 259
Zugriffsschlüssel-ID 180
Zusammenfassungsvektor 272