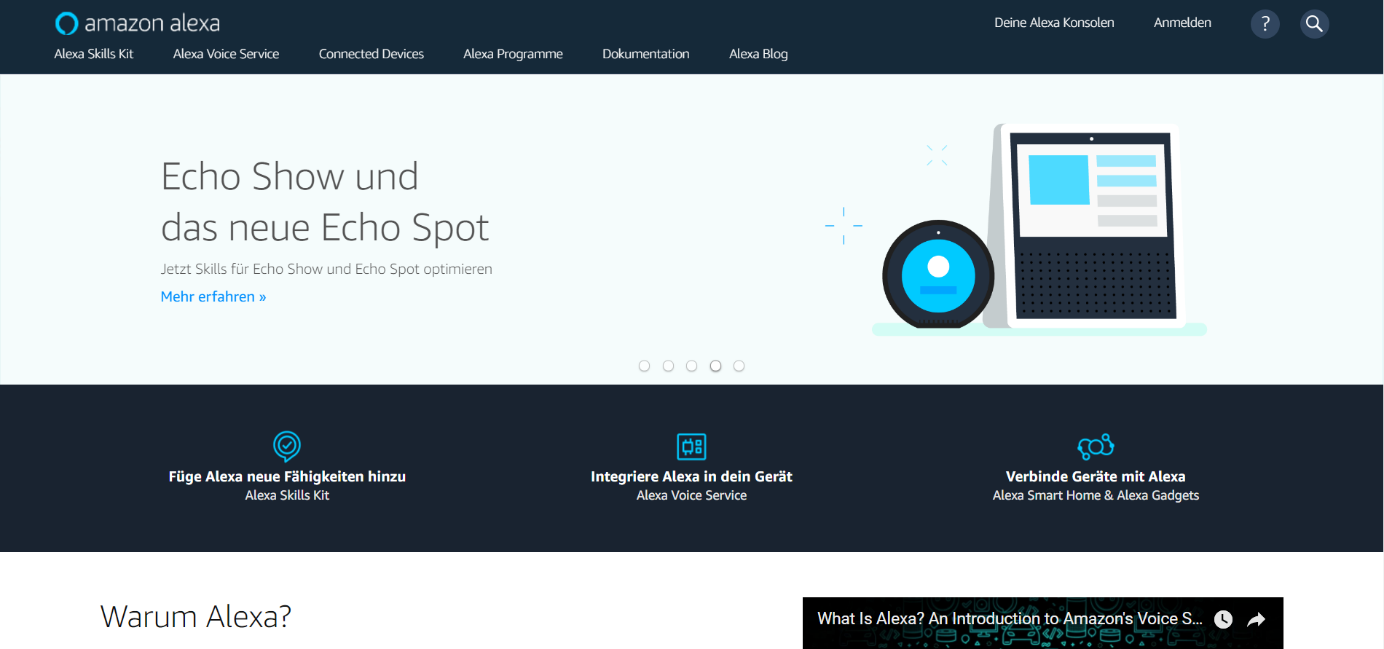
# Der Anfang

Kleine Vorwarnung! Falls keine Kreditkarte zur Verfügung steht KANN **KEIN** Skill entwickelt werden. Somit ist diese unabdingbar.  
Als aller erstes muss ein [neues Amazon Konto](https://www.amazon.de/ap/register?_encoding=UTF8&openid.assoc_handle=deflex&openid.claimed_id=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0%2Fidentifier_select&openid.identity=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0%2Fidentifier_select&openid.mode=checkid_setup&openid.ns=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0&openid.ns.pape=http%3A%2F%2Fspecs.openid.net%2Fextensions%2Fpape%2F1.0&openid.pape.max_auth_age=0&openid.return_to=https%3A%2F%2Fwww.amazon.de%2Fgp%2Fyourstore%2Fhome%3Fie%3DUTF8%26ref_%3Dnav_newcust) erstellt werden, falls noch keins existiert. Danach muss dieses zu einem Developer Konto „umgewandelt“ werden. Einfach auf die Homepage von [amazondeveloper](https://developer.amazon.com/de/) gehen und sich mit seinem bestehenden Amazon Konto anmelden.

# Den ersten Skill anlegen

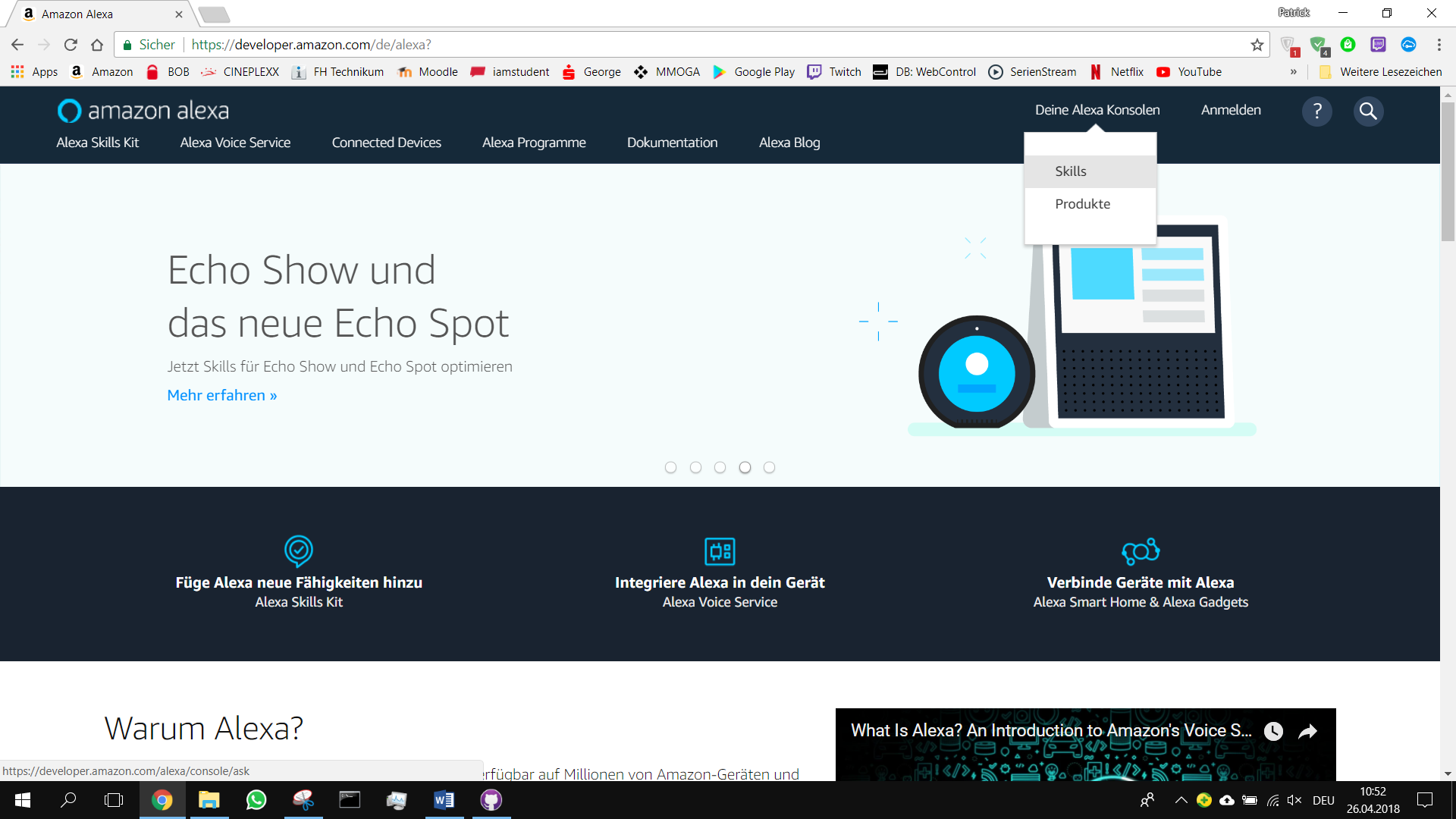
Ist man auf der Developer Seite von Amazon, muss zunächst auf „[amazon alexa](https://developer.amazon.com/de/alexa)“ geklickt werden um zu der Skill Entwicklung zu gelangen.

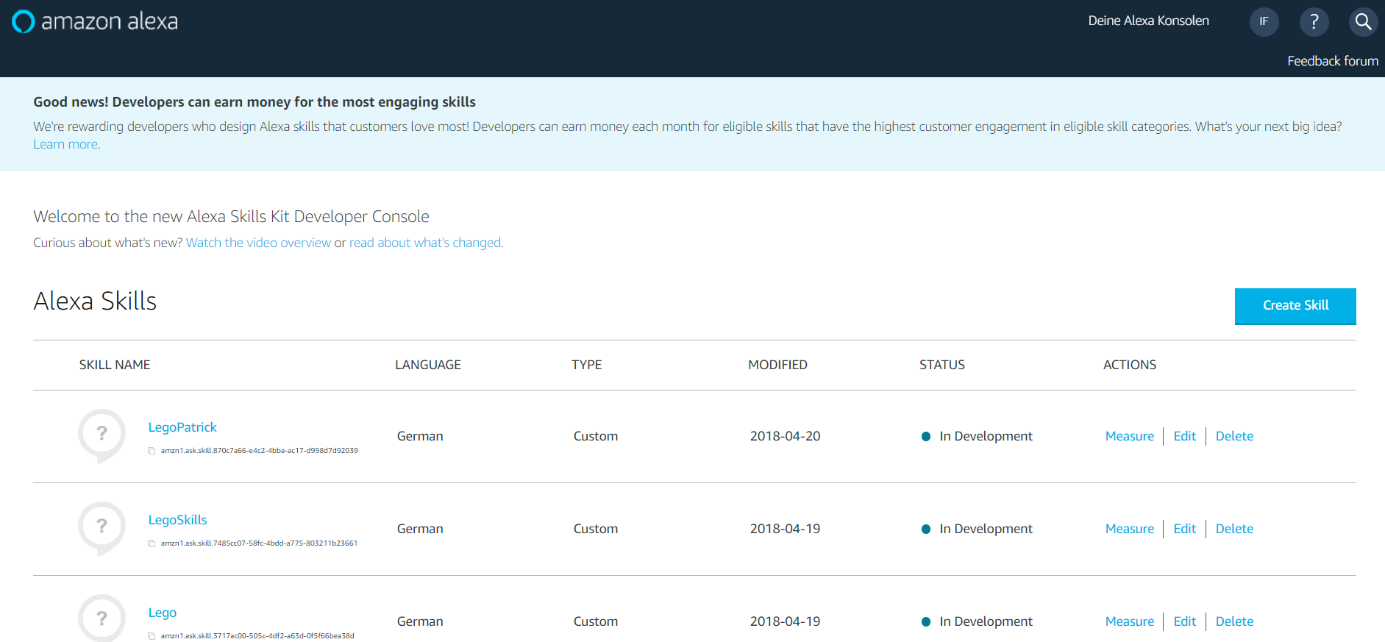


Das Design wurde von Amazon weitestgehend verändert, was Website und Amazon Skill Kit betrifft. Somit gibt es kaum Anleitungen zum aktuellen Design (Stand 26.014.2018).

Hier gelangt man zu den Skills.

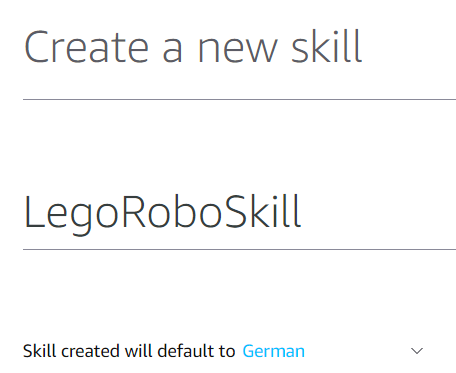
Hier erhält man Informationen zur Skill Entwicklung.



Ist man im Skill Kit so sollten hier eure Skins aufscheinen. Da ja noch kein Skill entwickelt wurde wird die Liste leer sein. In unserem Fall gibt es ja schon Skills, deshalb sieht man hier auch „LegoPatrick“, „LegoSkills“ und „Lego“.

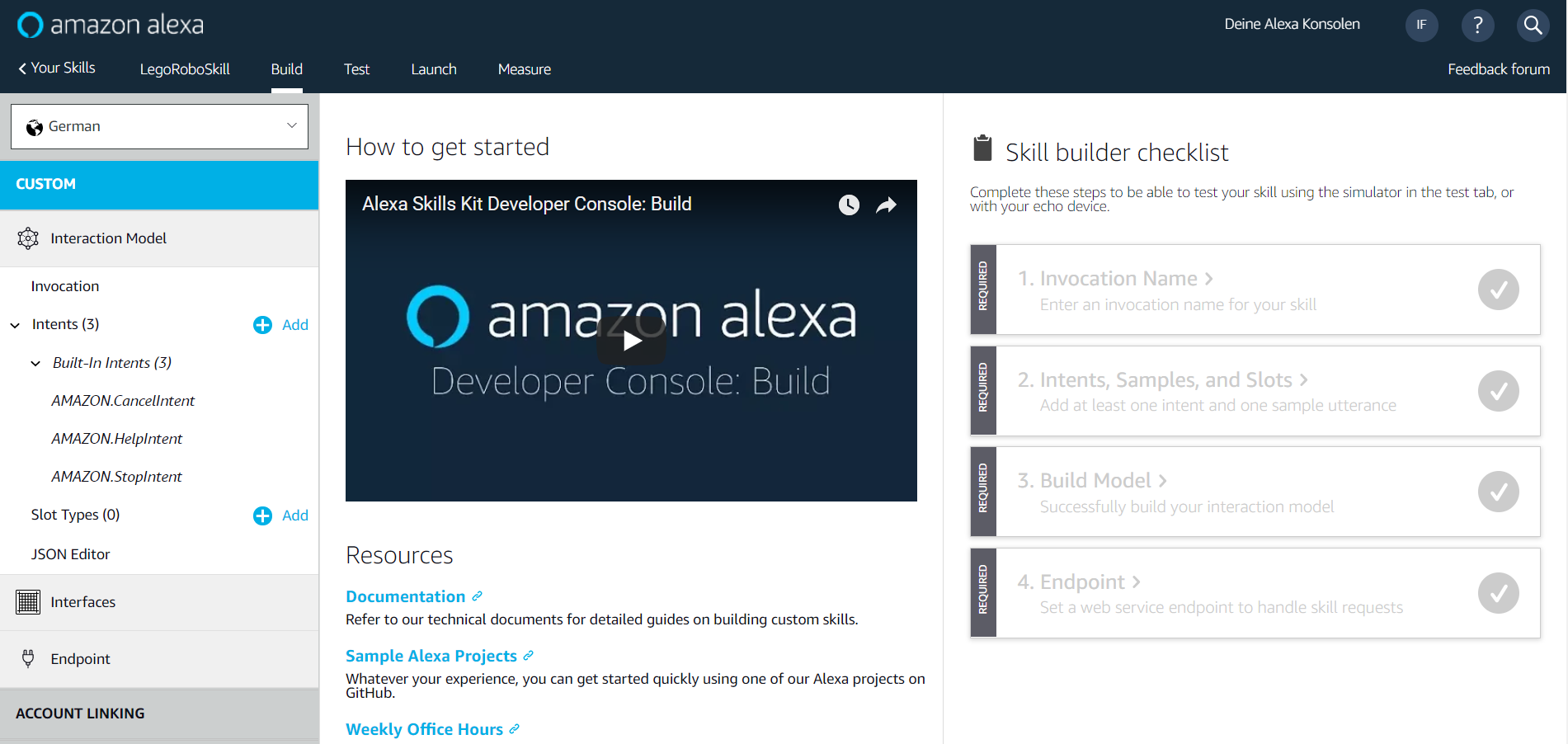
Um jetzt einen Skill zu erstellen klickt man auf „Create Skill“.

Man wird aufgefordert einen Namen einzugeben & die Sprache muss natürlich auch ausgewählt werden (in unserem Fall – German). Wir entscheiden uns für den Namen „LegoRoboSkill“.

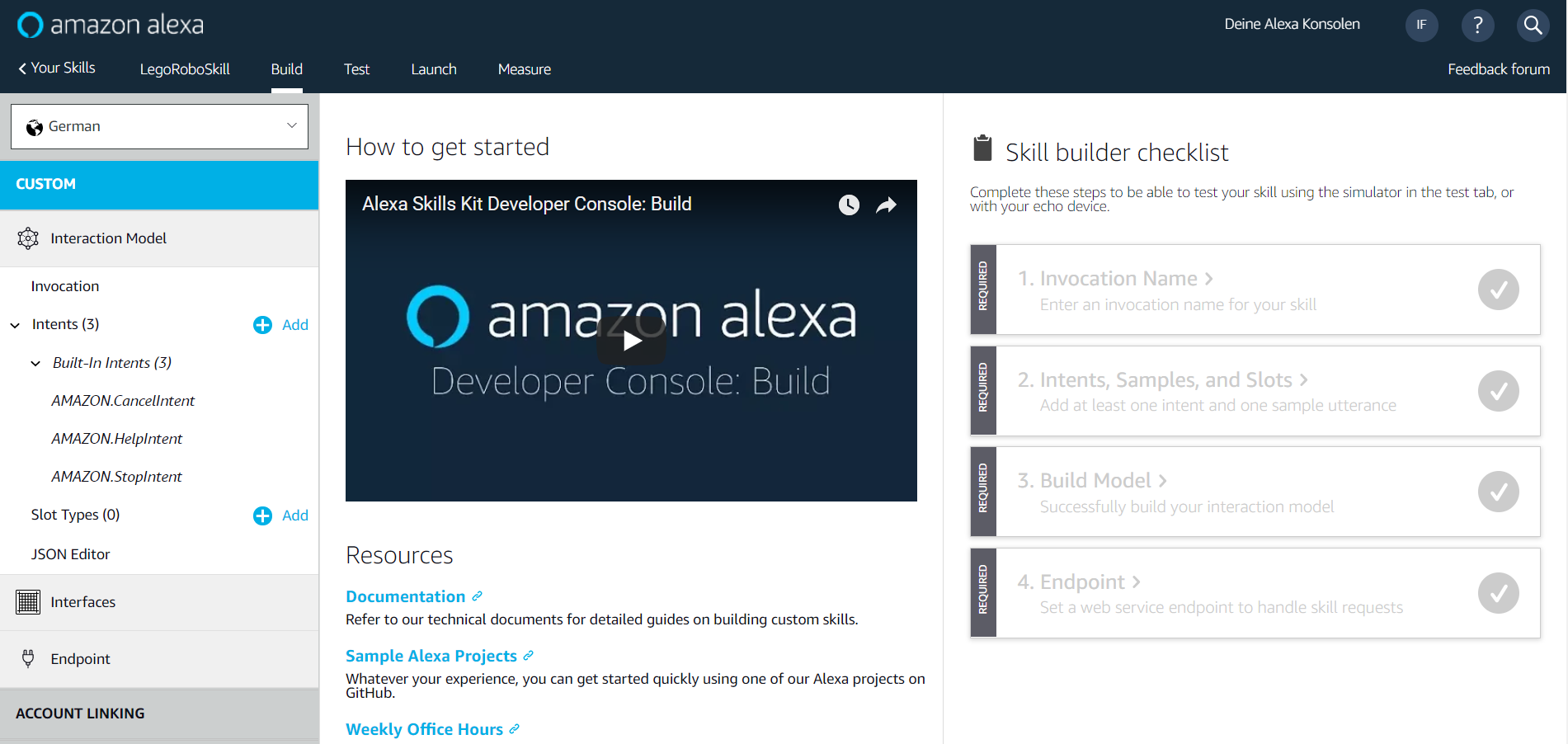


Im nächsten Schritt wählen wir aus, wofür unser Skill entwickelt werden soll. Da wir kein Smart Home oder dergleichen, sondern einen komplett eignen Skill entwickeln wollen muss „custom“ ausgewählt werden. Danach wird der Skill auch schon erstellt.

# Das Skill Kit Interface

Wie vorhin bereits erwähnt hat Amazon seine Benutzeroberfläche komplett überarbeitet und sieht nun folgendermaßen aus:

Zuerst wirkt die Seite sehr verwirrend, ist sie aber gar nicht. Sie ist eingeteilt in:  
  
*Build* (hier wird der aktuelle Skill entwickelt und ist auch der Reiter in dem wir uns am meisten bzw. fast immer aufhalten),   
  
*Test* (hier kann der Skill getestet werden, ist aber erst in ein paar Schritten wirklich nützlich, da zuerst ein Endpunkt zu AWS angelegt werden muss),   
  
*Lauch* & *Measure* sind nur wichtig falls der Skill veröffentlicht und allen zur Verfügung gestellt werden soll und da wir dies nicht vor haben ist es auch nicht wichtig.

Zurück zu dem Reiter ***Build***. Ich gebe euch einen kleinen Überblick wie die Seite aufgebaut ist, aber im Punkt 4, wird alles genau im Detail erklärt.

Wie der Name schon verrät werden hier die Endpoints hinzugefügt. (z.B.: AWS-Lambda)

Hier können Änderungen vorgenommen werden. (für uns unwichtig)

Die JSON Datei des Skills, welche auch bearbeitet werden kann.

Dadurch werden Slot Types(eine Art Variablen mit versch. Werten) hinzugefügt.

Hier können so genannte Intents(Befehle) hinzugefügt werden.

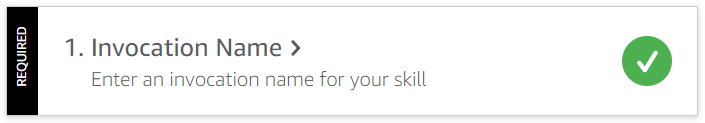
Hier wird der Invocation Name(Aufrufname) eingetragen.

# Den ersten Skill entwickeln

Auf der rechten Seite sieht man die **Skill Builder Checklist**, welche wir jetzt abarbeiten werden, um unseren Skill fertig stellen zu können.  
Als erster Punkt steht auf der Liste der Invocation Name. Das ist nichts anderes als der Name der Alexa gesagt werden muss, um das Programm bzw. den Skill zu starten. Auf der linken Seite gibt es den Punkt „Invocation“ und wenn wir darauf klicken werden wir aufgefordert diesen Namen eben einzugeben. In unserem Fall geben wir „lego skill“ ein. Der Name sollte leicht zu verstehen sein, damit Alexa es eindeutig identifizieren kann.  
Jedes Mal, wenn wir eine Änderung vornehmen sollte gespeichert werden, unter „Save Model“.

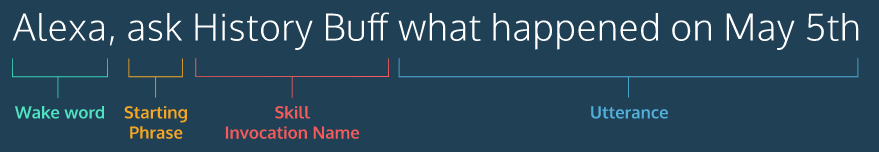


Gehen wir zurück sollte bei der Checklist ein grüner Haken sein:

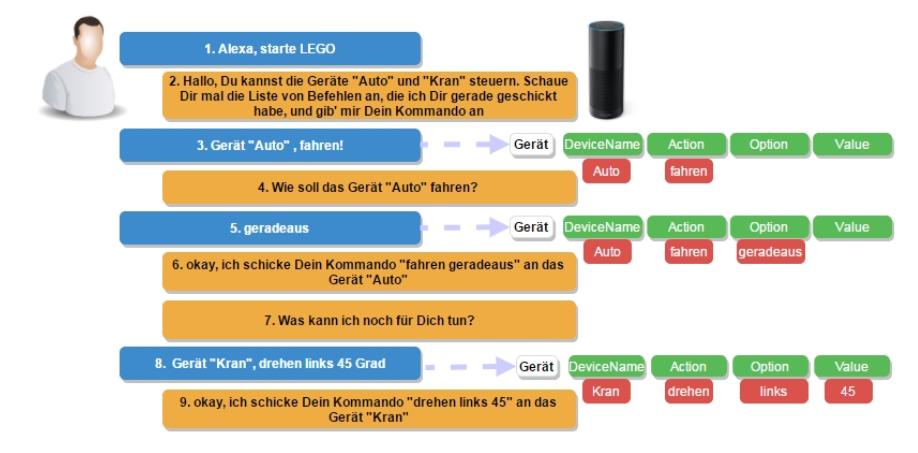


Als nächstes sollen wir Intents bzw. Utterances und Slots erstellen. Jetzt ist aber die Frage was sind diese Intents und Slots?  
Wiedermal klingt es komplizierter als es eigentlich ist.   
Intents oder Utterances sind sozusagen Befehle die Alexa von uns als Person erwartet und Slots sind Variablen die mehrere Werte haben können und diese wir ihnen auch zuweisen. Bevor man einen Skill entwickelt solle man sicher sein, was Alexa für mögliche Befehle erwarten kann, da dies wichtig ist. Am besten wir machen uns das Prinzip „learning per doing“ zu nutze.

Wir haben uns nach der Anleitung von [Vadim Kantor und Boris Kantor](https://jaxenter.de/alexa-skills-entwickeln-64751) orientiert.

Unser Alexa soll zwei Arten von Befehle verarbeiten können und zwar „Gerät Roboter fahre gerade aus“ und „Gerät Roboter drehe um 45 Grad“. Das bedeutet wir brauchen 2 Utterances und zwar einen ohne einer Grad Angabe und einen mit. Hier ein Beispiel um es zu veranschaulichen:

Hier wird der Skill mit dem Namen „History Buff“ aufgerufen und ein Utterance erwartet. Und zwar wird jetzt der Utterance „what happend on may 5th“ aufgerufen und an AWS-Lambda weitergegeben. Später dazu mehr.

Noch ein Beispiel, wie der Skill mit unseren Lego Robotern aussehen könnte:

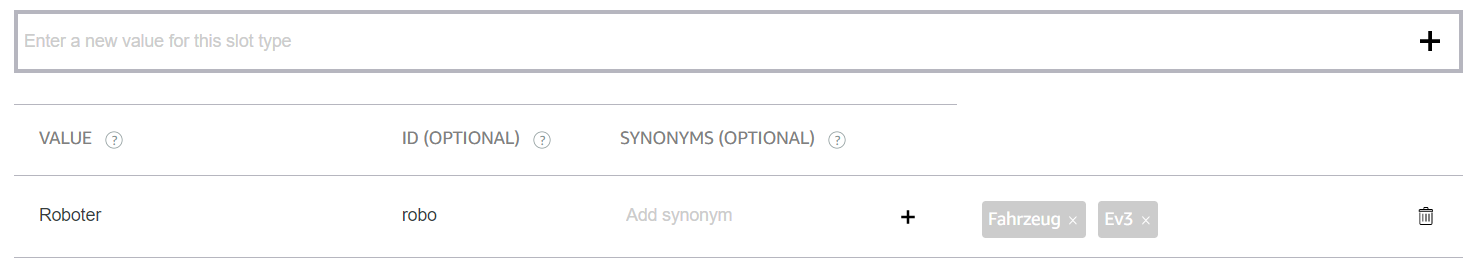
Hier sieht man auch schön wie Slot Types funktionieren. Nehmen wir unser obiges Beispiel von dem Utterance her: „Gerät Roboter fahre gerade aus“. Mit *Gerät* wird einfach unser Utterance gestartet. Danach wird der Slot Type „DeviceName“ erwartet welcher vorher definiert werden muss. Deshalb ist es wichtig vorher zu wissen wie der Alexa Skill aufgebaut werden soll. Zurück zum *DeviceName.* Der Slot Typ sollte hier auf jeden Fall *Roboter* gespeichert haben (sonst erkennt er diesen Befehl nicht) und wird ausgewählt und bei *Action,* wird *fahren* ausgewählt, usw..

## Slot Types

So da jetzt nicht mehr so unklar ist was Slot-Types sind, können wir diese nun erstellen.

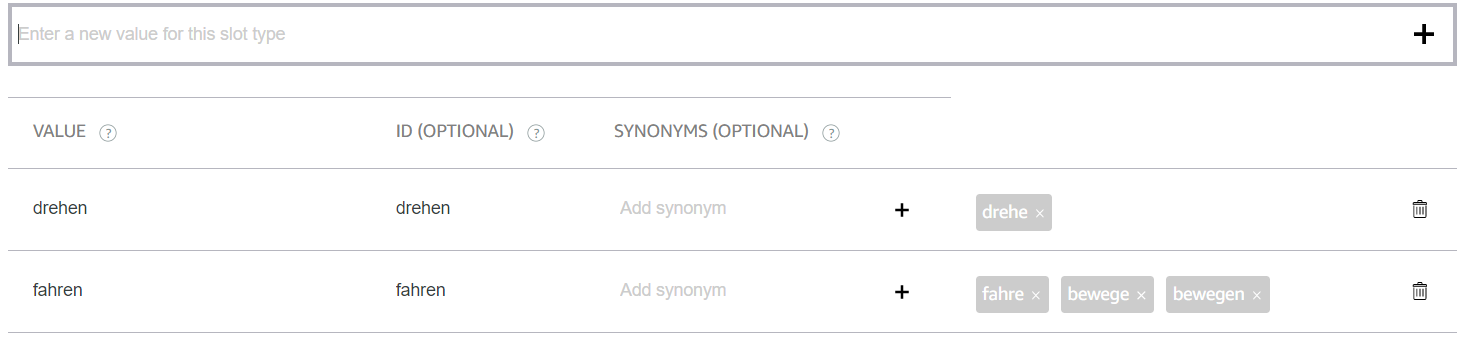
Und zwar brauchen wir 3 Slot Types. Einen für unseren Gerätenamen (*DeviceName*), einen für die Funktion was er tun soll (*Action*) und einen wie er diese Funktion ausführen soll (*Option*). Für die Grad Angabe kann der vorgefertigte Type -> *Amazon.Number* verwendet werden.

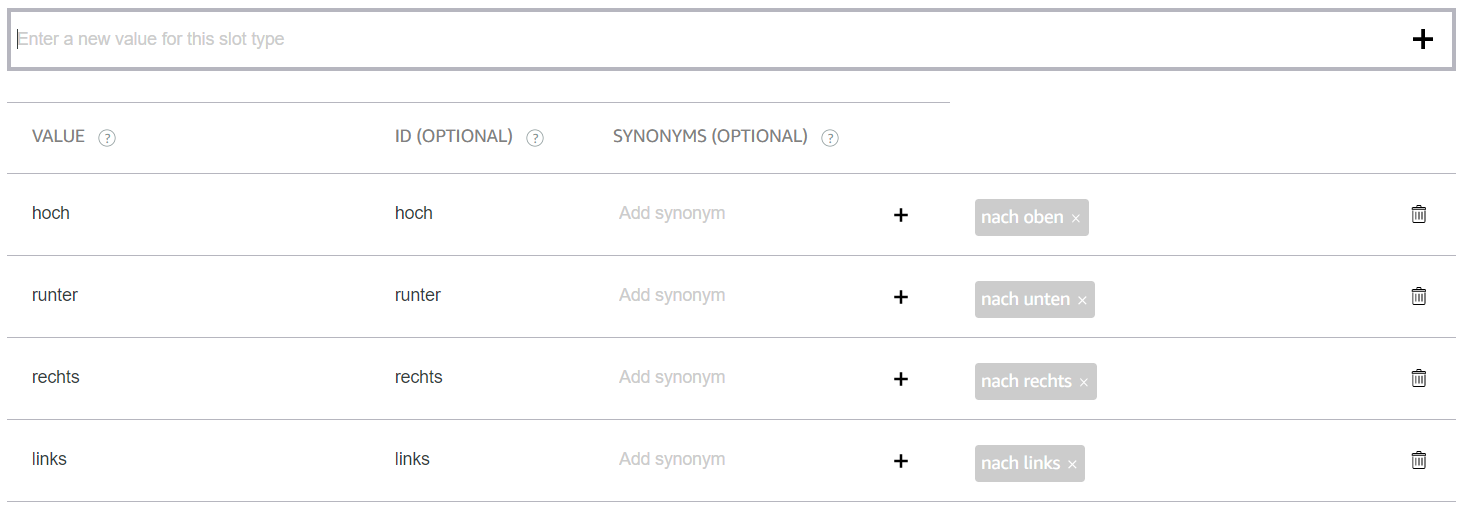
Ist *Device\_Name\_Type* erstellt fügen wir noch eine ID und Synonyme hinzu. Es sollte dann folgendermaßen aussehen:



Unser Roboter kann also über *Roboter*, *Fahrzeug* und *Ev3* aufgerufen werden.

Das gleiche machen wir nun für *Action\_Type*:

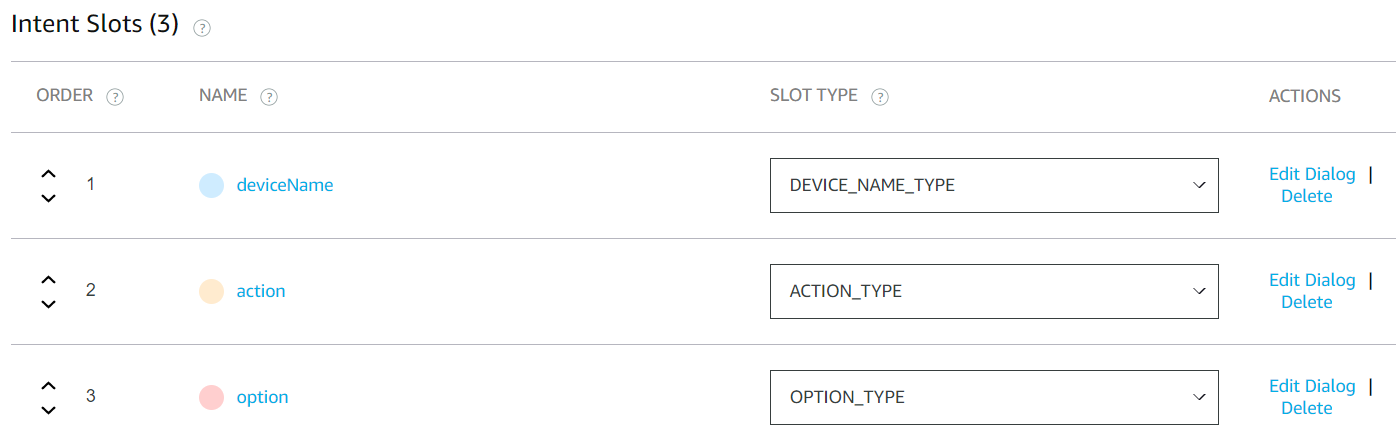


und *Option\_Type*:

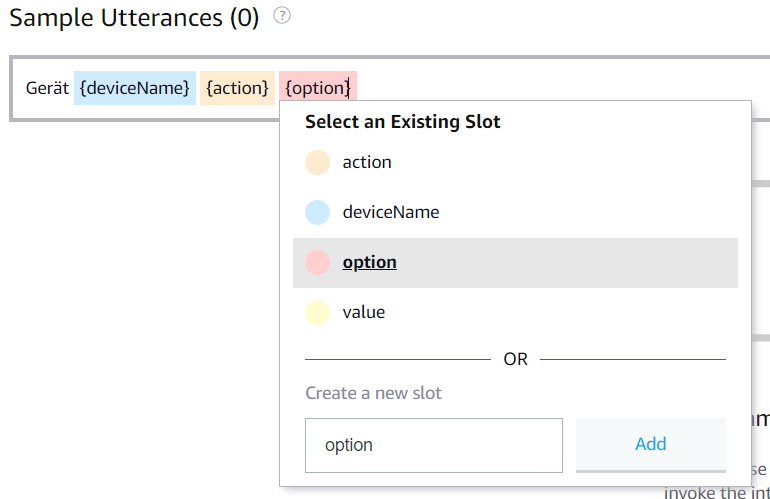
## Intents und Utterance

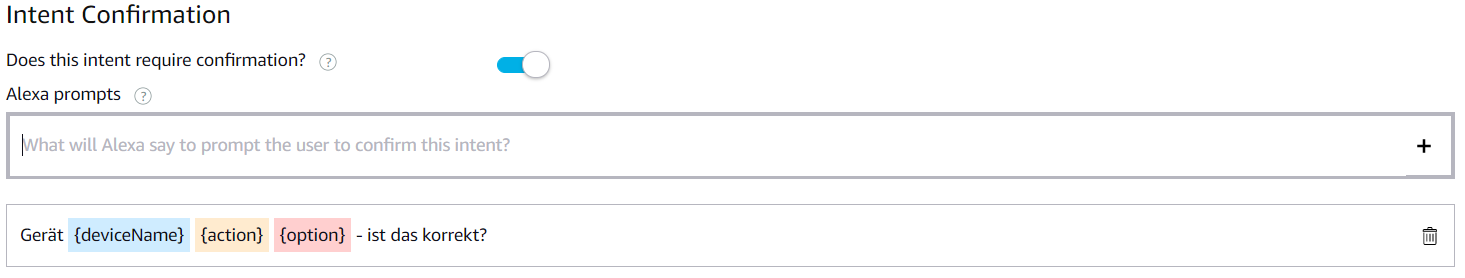
Wir fügen nun unseren Intent ohne Gradangabe hinzu und nennen diesen „CommandIntent“.

Als erstes müssen wir unsere Slot-Types in die Intent Slots eintragen. Wir nennen dieses Slots so, wie wir sie in unserem Utterance aufrufen wollen.



Jetzt können wir unser Utterance erstellen. Die Intent Slots werden mit einer geschwungenen Klammer aufgerufen:



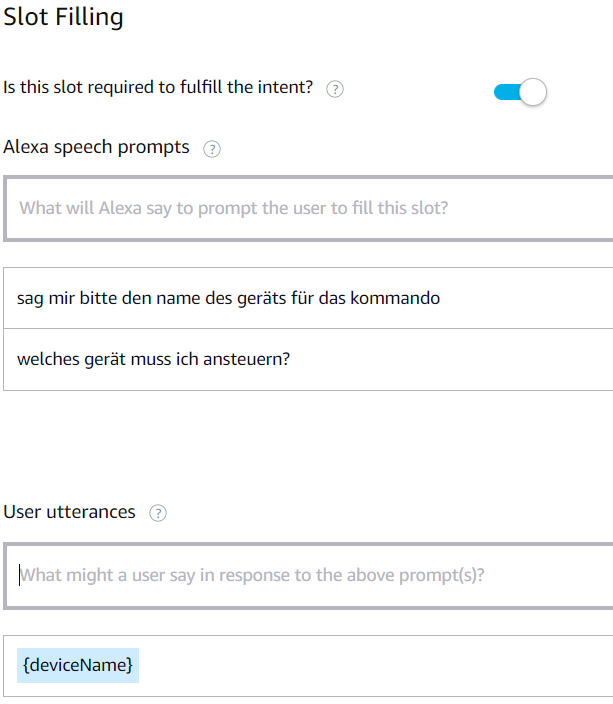
Jetzt lassen wir Alexa den verstandenen Input wiederholen und vom User bestätigen. Das Funktioniert über den Intent Confirmation.

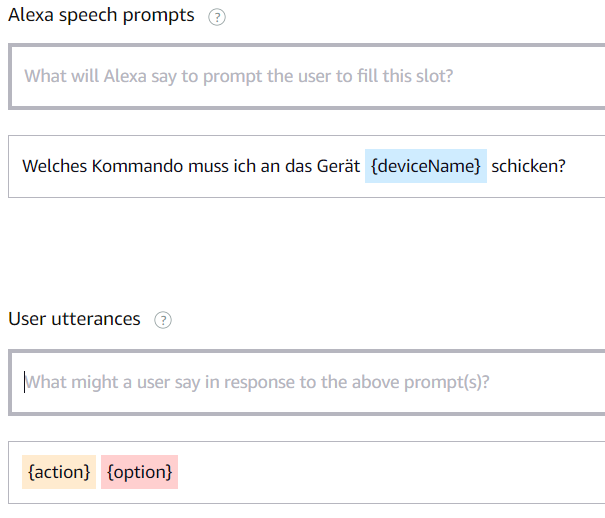
Natürlich kann es sein das ein User etwas falsch sagt oder vergisst. Somit müssen wir manche Slot Types auf required setzten und dazu weitere Einstellungen treffen.

Beginnen wir mit den *DeviceName*:

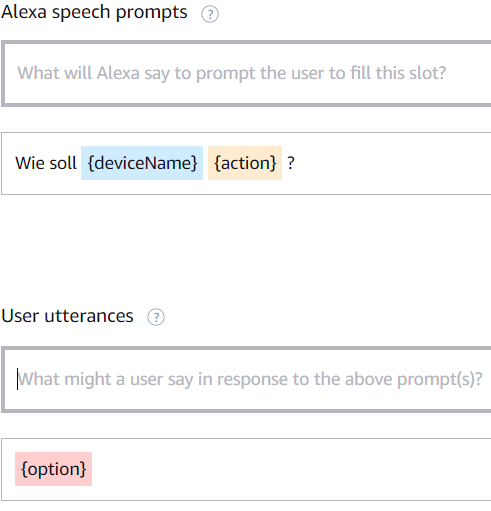
Alexa soll ja darauf regieren, wenn der User etwas falsch getan hat. Somit soll Sie ihm mitteilen an welches Gerät, sprich den Gerätenamen wir etwas übermitteln wollen.

Außerdem soll die Antwort des Users wieder in einen Slot Type gespeichert werden. Dies sollte dann folgendermaßen aussehen:

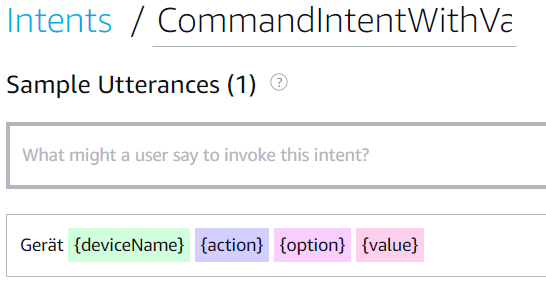


Das gleiche wiedermal für *Action*:

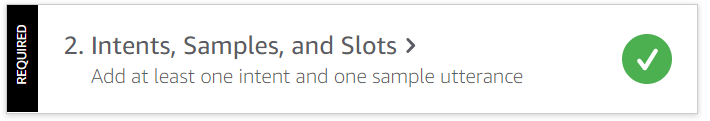
Und *Option*:



In diesem Utterance können nur Befehle entgegen genommen werden ohne eine präzise Drehung mit einer Grad Angabe. Somit muss der Schritt 4.2. noch einmal wiederholt werden nur mit einem Utterance der auch eine Gradangabe erwartet. Alexa erkennt dieses und weiß welchen Utterance sie verwenden soll.



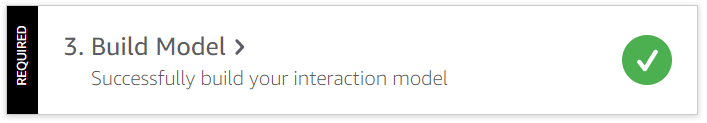
Da wir mit Intents, Utterance und Slot Types fertig sind sollte ein grünes Häkchen in der Checklist sein.



Der nächste Punkt auf der Checklist ist einfach das Projekt mal zu builden, über den Button „Build Model“.



Sind keine Fehler passiert sollte auch hier ein grüner Haken sein.



## Endpoints

Bei Endpoints wird eine Verbindung zum AWS von Amazon hergestellt. Um genau zu sein zu [Lambda](https://aws.amazon.com/de/lambda/features/).

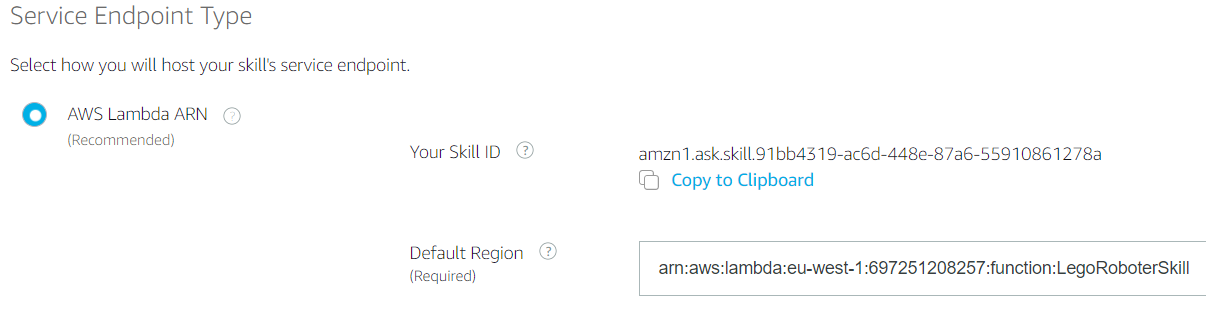
Um zu verstehen was Lambda genau ist, zitiere ich hier dessen Beschreibung:

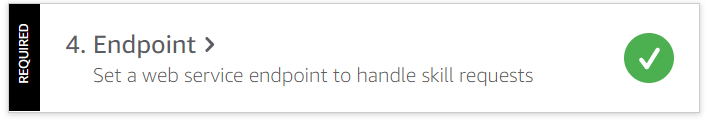
„AWS Lambda ist ein serverloser Datenverarbeitungsservice, der Ihren Code beim Eintreten bestimmter Ereignisse ausführt und automatisch für Sie die zugrunde liegenden Datenverarbeitungsressourcen verwaltet. Mit AWS Lambda können Sie andere AWS-Services mit benutzerdefinierter Logik erweitern oder Ihre eigenen Backend-Services erstellen und im AWS-Umfang und mit AWS-Leistung und -Sicherheit betreiben. AWS Lambda kann Code automatisch als Reaktion auf mehrere Ereignisse wie HTTP-Anforderungen über Amazon API Gateway, Änderungen von Objekten in Amazon S3-Buckets, Aktualisierungen von Tabellen in Amazon DynamoDB oder Zustandsübergänge in AWS Step Functions ausführen.

Lambda führt Ihren Code automatisch auf hochverfügbarer Datenverarbeitungsinfrastruktur aus und erledigt die gesamte Administration der Datenverarbeitungsressourcen, einschließlich Server- und Betriebssystemwartung, Kapazitätsbereitstellung und automatischer Skalierung, Code- und Sicherheitspatch-Bereitstellung sowie Code-Überwachung und -Protokollierung. Sie müssen nur den Code zur Verfügung stellen.“

DAVID DOKU EINFÜGEN

Hat man die Lambda Funktion fertig muss nun dessen ARN bei Default-Region eingefügt werden.

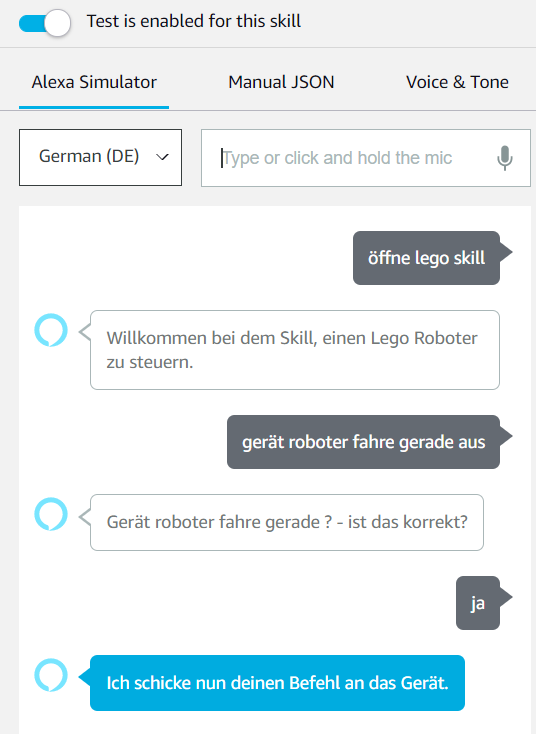




Kann der Endpoint gespeichert werden sind keine Fehler aufgetreten und es kann getestet werden.

Grundlegend gibt es 2 gute Möglichkeiten seinen Skill zu testen. Einerseits [Echosim.io](https://echosim.io/welcome) oder direkt in dem Alexa Skill Kit.

Wir Testen in Alexa Skill Kit, da es uns mehr Informationen gibt und auch über einen Text angesprochen werden kann.   
Ein Skill wird mit „öffne“ oder „starte“ + Invocation Name gestartet.



Hier sieht man wie ein möglicher Aufruf aussehen könnte.