

amazon echo



EINFÜHRUNG IN DIE
ALEXA SKILL ENTWICKLUNG

Übersicht

- ▷ Teil 1: Hallo Alexa
- ▷ Teil 2: Slots, Slot Types und Utterances
- ▷ Teil 3: Sessions und Voice User Interfaces
- ▷ Teil 4: Ausblick



Teil 3: Sessions und Voice User Interfaces

Übersicht Teil 3

1. Lückentext Skill bauen
2. Wieso Sessions?
3. Praktischer Teil
4. Service auf AWS Lambda bereitstellen (Demo)
5. Alexa Skill Interface konfigurieren (Demo)



Alexa, starte Lückentext.

Nenne mir ein Verb in
Vergangenheitsform.



Aß.

Nenne mir einen Namen.



Marco Polo.

1.

Lückentext Skill bauen

Nenne mir ein Substantiv.



Hilfe.

Nenne mir ein Substantiv um es zum Lückentext hinzuzufügen. Ein Substantiv ist ein ein Wort, das einen Menschen, eine Sache, ein Tier oder Ähnliches bezeichnet: ein Polizist, ein Ei oder ein Affe. Wie lautet dein Substantiv?



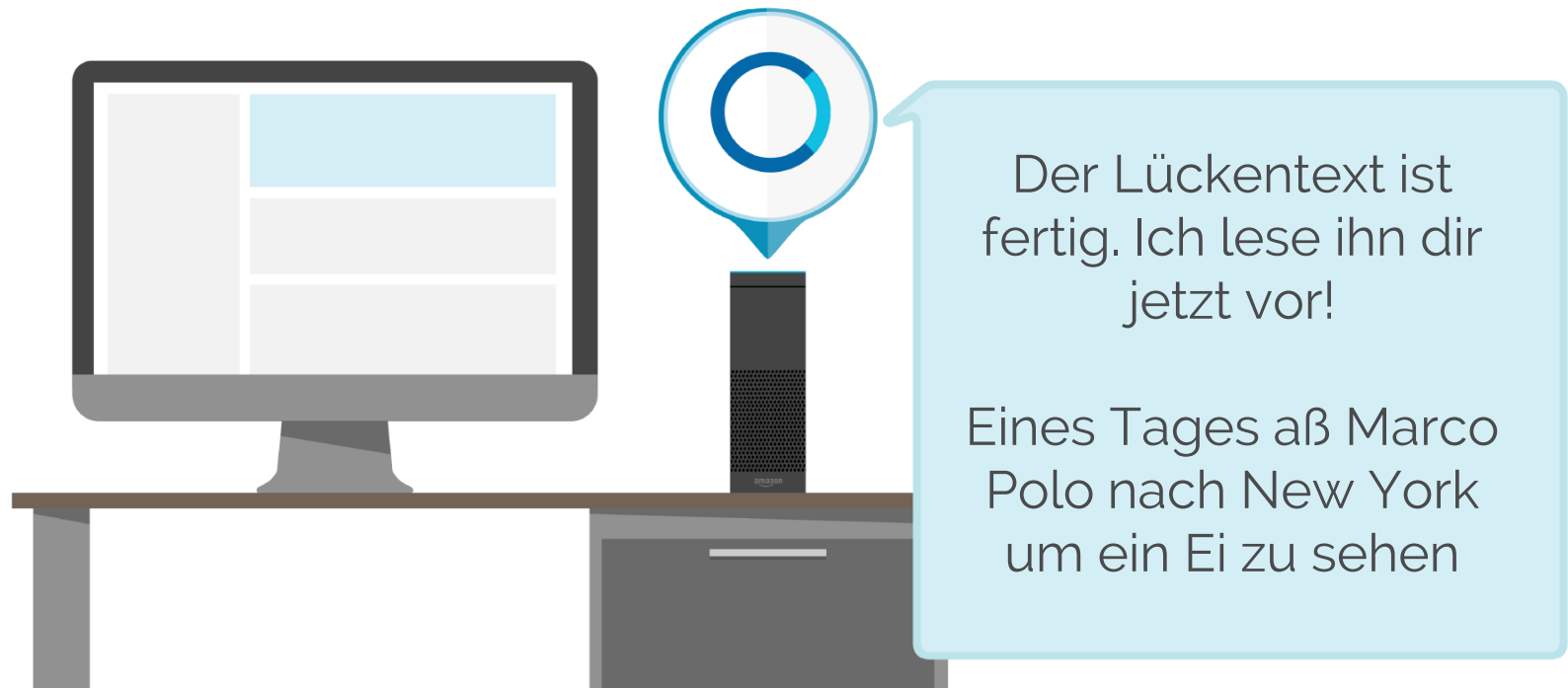
Ein Ei.

Der Lückentext ist fertig. Ich lese ihn
dir jetzt vor!

Eines Tages aß Marco Polo nach
New York um ein Ei zu sehen



Karten in der Alexa App



2.

Wieso Sessions?



WTF?

Bitte nenne mir deinen
Vornamen, deinen
Nachnamen, deinen Twitter
Namen und deine
Telefonnummer.



3.

Praktischer Teil

Lückentext Launch Handler definieren

```
"use strict";
module.change_code = 1;
var _ = require("lodash");
var Skill = require("alexa-app");
var skillService = new Skill.app("lueckentextbuilder");
var LueckentextHelper = require("../lueckentext_helper");

skillService.launch(function(request, response) {
    var prompt = "Willkommen zu Lückentext. "
        + "Um einen neuen Lückentext zu erstellen, sage
    erstelle einen Lückentext. ";
    response.say(prompt).shouldEndSession(false);
});

module.exports = skillService;
```

Invocation Typen

Kein Intent

Der Benutzer fragt Alexa, ohne weitere Details anzugeben.

Beispiel: *Alexa öffne Lückentext.*

Unvollständiger Intent

Der Benutzer drückt nur einen Teil aus, was für ihn erforderlich ist.

Beispiel: *Alexa, frage Bahnauskunft nach Abfahrtszeiten.*

(benötigt genauere Informationen)

Voller Intent

Der Benutzer drückt alles aus, was erforderlich ist, um seine Anfrage in einer einzigen Äußerung abzuschließen.

Beispiel: *Alexa, frage Bahnauskunft nach Abfahrtszeiten für Wiesbaden Hauptbahnhof.*



Alexa, starte Lückentext.

Willkommen zu Lückentext.
Um einen neuen Lückentext
zu erstellen, sage erstelle
einen Lückentext.



AMAZON.HelpIntent Handler hinzufügen

```
[...]
skillService.launch(function(request, response) {
  var prompt = "Willkommen zu Lückentext. "
    + "Um einen neuen Lückentext zu erstellen, sage erstelle einen
Lückentext. "; response.say(prompt).shouldEndSession(false);
});

skillService.intent("AMAZON.HelpIntent", {},
function(request, response) {
  var lueckentextHelper = getLueckentextHelper(request);
  var help = "Willkommen zu Lückentext. "
    + "Um einen neuen Lückentext zu erstellen, sage erstelle einen
Lückentext. "
    + "Du kannst auch stopp oder abbrechen sagen um den Skill zu beenden.";
  if (lueckentextHelper.started) {
    help = lueckentextHelper.getStep().help;
  }
  response.say(help).shouldEndSession(false);
});

module.exports = skillService;
```


Built-in Intents¹

Intent	Erlaubte Äußerung	Einsatzzweck
AMAZON.CancelIntent	abbrechen, abbreche, vergiss es	Bricht entweder eine Transaktion oder Aufgabe ab (aber bleibt noch im Skill) oder schließt den Skill vollständig.
AMAZON.HelpIntent	Hilfe (bitte), Hilf mir (bitte), Kannst du mir helfen (bitte)	Gibt Auskunft darüber, wie man den Skill benutzt
AMAZON.NoIntent	Nein, Nein danke	Negative Antwort auf eine Ja/Nein Frage.
AMAZON.StopIntent	stopp, hör endlich auf, aufhören	Aktion wird gestoppt.
AMAZON.YesIntent	Ja, Ja bitte	Positive Antwort auf Ja/Nein Frage.

AMAZON.StopIntent und AMAZON.CancelIntent hinzufügen

[...]

```
var cancelIntentFunction = function(request, response) {  
    response.say("Auf Wiedersehen!").shouldEndSession(true);  
};
```

```
skillService.intent("AMAZON.CancelIntent", {},  
cancelIntentFunction);  
skillService.intent("AMAZON.StopIntent", {},  
cancelIntentFunction);
```

```
skillService.launch(function(request, response) {  
    var prompt = "Willkommen zu Lückentext. „
```

[...]

LueckentextIntent Handler hinzufügen

```
[...]
skillService.intent("AMAZON.HelpIntent", {},
    function(request, response) {
        ...
    });

skillService.intent("lueckentextIntent", {
    "slots": {
        "STEPVALUE": "STEPVALUES"
    },
    "utterances": ["{neuer|starte|erstelle|beginne|baue} {einen|den}
Lückentext", "{-|STEPVALUE}"]
},
    function(request, response) {
        //Lückentext Funktionalität!
    }
);
module.exports = skillService;
```

Step Values speichern

[...]

```
function(request, response) {  
  //Lückentext Funktionalität!  
  var stepValue = request.slot("STEPVALUE");  
  var lueckentextHelper = new LueckentextHelper();  
  lueckentextHelper.started = true;  
  if (stepValue !== undefined) {  
    lueckentextHelper.getStep().value = stepValue;  
  }  
  if (lueckentextHelper.completed()) {  
    var completedLueckentext = lueckentextHelper.buildLueckentext();  
    response.card(lueckentextHelper.currentLueckentext().title, completedLueckentext);  
    response.say("Der Lückentext ist fertig! Ich lese ihn dir jetzt vor. " +  
completedLueckentext);  
    response.shouldEndSession(true);  
  } else {  
    response.say("Nenne mir " + lueckentextHelper.getPrompt());  
    response.reprompt("Ich habe leider nichts gehört. Nenne mir " +  
lueckentextHelper.getPrompt() + " um fortzufahren.");  
    response.shouldEndSession(false);  
  }  
}
```

[...]

Den Session Status speichern und abrufen

```
[...]
var LueckentextHelper = require("./lueckentext_helper");
var LUECKENTEXT_BUILDER_SESSION_KEY = "lueckentext_builder";
var getLueckentextHelper = function(request) {
    var lueckentextHelperData =
request.session(LUECKENTEXT_BUILDER_SESSION_KEY);
    if (lueckentextHelperData === undefined) {
        lueckentextHelperData = {};
    }
    return new LueckentextHelper(lueckentextHelperData);
};
var cancelIntentFunction = function(request, response) {
[...]
```

Session Status abrufen

```
[...]
function(request, response) {
  var stepValue = request.slot("STEPVALUE");
var lueckentextHelper = new LueckentextHelper();
  var lueckentextHelper = getLueckentextHelper(request);
  lueckentextHelper.started = true;
  if (stepValue !== undefined) {
    lueckentextHelper.getStep().value = stepValue;
  }
  if (lueckentextHelper.completed()) {
    var completedLueckentext = lueckentextHelper.buildLueckentext();
    response.card(lueckentextHelper.currentLueckentext().title, completedLueckentext);
    response.say("Der Lückentext ist fertig! Ich lese ihn dir jetzt vor. " +
completedLueckentext);
    response.shouldEndSession(true);
  } else {
    if (stepValue !== undefined) {
      lueckentextHelper.currentStep++;
    }
    response.say("Nenne mir " + lueckentextHelper.getPrompt());
    response.reprompt("Ich habe leider nichts gehört. Nenne mir " +
lueckentextHelper.getPrompt() + " um fortzufahren.");
    response.shouldEndSession(false);
  }
  response.session(LUECKENTEXT_BUILDER_SESSION_KEY, lueckentextHelper);
}
);
[...]
```

4.

Service auf AWS Lambda
bereitstellen (Demo)

5.

Alexa Skill Interface konfigurieren (Demo)

Mittagspause!