

Untitled2

December 16, 2021

```
[1]: # -*- coding: utf-8 -*-

# This sample demonstrates handling intents from an Alexa skill using the Alexa
↳ Skills Kit SDK for Python.
# Please visit https://alexa.design/cookbook for additional examples on
↳ implementing slots, dialog management,
# session persistence, api calls, and more.
# This sample is built using the handler classes approach in skill builder.
import logging
import ask_sdk_core.utils as ask_utils

from ask_sdk_core.skill_builder import SkillBuilder
from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.dispatch_components import AbstractExceptionHandler
from ask_sdk_core.handler_input import HandlerInput

from ask_sdk_model import Response

logger = logging.getLogger(__name__)
logger.setLevel(logging.INFO)

class LaunchRequestHandler(AbstractRequestHandler):
    """Handler for Skill Launch."""
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool

        return ask_utils.is_request_type("LaunchRequest")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Willkommen bei Skin Guide! Schön, dass du da bist! Hilf
↳ mir dich besser kennenzulernen. Wie heißt du?"

        return (
            handler_input.response_builder
                .speak(speak_output)
```

```

        .ask(speak_output)
        .response
    )

class HelloWorldIntentHandler(AbstractRequestHandler):
    """Handler for Hello World Intent."""
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.is_intent_name("HelloWorldIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Hello World!"

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

class CancelOrStopIntentHandler(AbstractRequestHandler):
    """Single handler for Cancel and Stop Intent."""
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return (ask_utils.is_intent_name("AMAZON.CancelIntent")(handler_input)
→or
                ask_utils.is_intent_name("AMAZON.StopIntent")(handler_input))

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Goodbye!"

        return (
            handler_input.response_builder
                .speak(speak_output)
                .response
        )

class SessionEndedRequestHandler(AbstractRequestHandler):
    """Handler for Session End."""
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool

```

```

        return ask_utils.is_request_type("SessionEndedRequest")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response

        # Any cleanup logic goes here.

        return handler_input.response_builder.response

class IntentReflectorHandler(AbstractRequestHandler):
    """The intent reflector is used for interaction model testing and debugging.
    It will simply repeat the intent the user said. You can create custom
    ↪handlers
    for your intents by defining them above, then also adding them to the
    ↪request
    handler chain below.
    """
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.is_request_type("IntentRequest")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        intent_name = ask_utils.get_intent_name(handler_input)
        speak_output = "You just triggered " + intent_name + "."

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
    ↪the user to respond")
                .response
        )

class CatchAllExceptionHandler(AbstractExceptionHandler):
    """Generic error handling to capture any syntax or routing errors. If you
    ↪receive an error
    stating the request handler chain is not found, you have not implemented a
    ↪handler for
    the intent being invoked or included it in the skill builder below.
    """
    def can_handle(self, handler_input, exception):
        # type: (HandlerInput, Exception) -> bool
        return True

```

```

def handle(self, handler_input, exception):
    # type: (HandlerInput, Exception) -> Response
    logger.error(exception, exc_info=True)

    speak_output = "Sorry, I had trouble doing what you asked. Please try_
↪again."

    return (
        handler_input.response_builder
            .speak(speak_output)
            .ask(speak_output)
            .response
    )

class HauttypBekanntHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
↪is_hauttyp_inhaltsstoffe("HauttypBekanntIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Schön das du es weißt. Welchen Hauttyp hast du {name}?_
↪Es gibt folgende Hauttypen: Normale Haut, fettige Haut, trockene Haut und_
↪Mischhaut."

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for_
↪the user to respond")
                .response
        )

class HauttypBekanntNormaleHautHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
↪is_hauttyp_bekannt_normale_haut("HauttypBekanntNormaleHautIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response

```

```

        speak_output = "Super! Für deine normale Haut kann ich dir folgende_
↳Inhaltsstoffe empfehlen:Polyhydroxysäure, weißer Tee, Squalan, Mandelsäure,_
↳Panthenol, Arganöl, Hyaluronsäure, Glykolsäure und Ceramide.Möchtest du eine_
↳passende DIY-Maske für deine normale Haut wissen?"

        return (
            handler_input.response_builder
                .speak(speak_output)
                .ask(speak_output)
                .response
        )

class HauttypBekanntFettigeHautHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
↳is_hauttyp_bekannt_fettige_haut("HauttypBekanntFettigeHautIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Super! Für deine fettige Haut kann ich folgende_
↳Inhaltsstoffe empfehlen:Salicylsäure, Niacinamid, Hagebutten Kernöl,_
↳Mandelsäure, Tonerde, Teebaumöl, Benzoylperoxid, Urea und Squalan.Möchtest_
↳du eine passende DIY-Maske für deine fettige Haut wissen?"

        return (
            handler_input.response_builder
                .speak(speak_output)
                .ask(speak_output)
                .response
        )

class HauttypBekanntTrockeneHautHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
↳is_hauttyp_bekannt_trockene_haut("HauttypBekanntTrockeneHautIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Super! Für deine trockene Haut kann ich dir folgende_
↳Inhaltsstoffe empfehlen:Milchsäure, Urea, Glykolsäure, Ceramide, Sheabutter,_
↳Hyaluronsäure, Vaseline und Arganöl.Möchtest du eine passende DIY-Maske für_
↳deine trockene Haut wissen?"

        return (

```

```

        handler_input.response_builder
            .speak(speak_output)
            .ask(speak_output)
            .response
    )

class HauttypBekanntMischhautHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_hauttyp_bekannt_mischhaut("HauttypBekanntMischhautIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Super! Für deine Mischhaut kann ich dir folgende
→Inhaltsstoffe empfehlen: Salicylsäure, Glycerin, Wildrosenöl, Hyaluronsäure,
→Niacinamid, Jojoba-Öl, Glykolsäure, Aloe Vera und Squalan. Möchtest du eine
→passende DIY-Maske für deine Mischhaut wissen?"

        return (
            handler_input.response_builder
                .speak(speak_output)
                .ask(speak_output)
                .response
        )

class HauttypUnbekanntHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_hauttyp_inhaltsstoffe("HauttypUnbekanntIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Okay, dann finden wir das gemeinsam heraus! Wird deine
→Haut im Laufe des Tages fettig, trocken, fettig und trocken oder unverändert?
→ "

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

```

```

class HauttypUnbekanntFettigeHautHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_hauttyp_bekannt_fettige_haut("HauttypUnbekanntFettigeHautIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Beobachtest Du auf deiner Haut Rötungen oder Juckreiz,
→Mitesser, alle drei zusammen oder keines dieser drei Antwortmöglichkeiten? "

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

class HauttypUnbekanntTrockeneHautHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_hauttyp_bekannt_trockene_haut("HauttypUnbekanntTrockeneHautIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Beobachtest Du auf deiner Haut Rötungen oder Juckreiz,
→Mitesser, alle drei zusammen oder keines dieser drei Antwortmöglichkeiten? "

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

class HauttypUnbekanntNormaleHautHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_normale_haut("HauttypUnbekanntNormaleHautIntent")(handler_input)

```

```

def handle(self, handler_input):
    # type: (HandlerInput) -> Response
    speak_output = "Beobachtest Du auf deiner Haut Rötungen oder Juckreiz,␣
↳ Mitesser, alle drei zusammen oder keines dieser drei Antwortmöglichkeiten? "

    return (
        handler_input.response_builder
            .speak(speak_output)
            # .ask("add a reprompt if you want to keep the session open for␣
↳ the user to respond")
            .response
    )

class HauttypUnbekanntMischhautHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
↳ is_mischhaut_antwort_eins("HauttypUnbekanntNormaleHautIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Beobachtest Du auf deiner Haut Rötungen oder Juckreiz,␣
↳ Mitesser, alle drei zusammen oder keines dieser drei Antwortmöglichkeiten? "

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for␣
↳ the user to respond")
                .response
        )

class HauttypUnbekanntFettigeHautZweiHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
↳ is_fettige_haut_antwort_zwei("HauttypUnbekanntFettigeHautZweiIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Würdest du dein Hautbild eher rissig und spröde oder␣
↳ glänzend oder rosig und zart einschätzen?"

        return (
            handler_input.response_builder
                .speak(speak_output)

```



```

        # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
        .response
    )

class HauttypUnbekanntTrockeneHautZweiHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_trockene_haut_antwort_zwei("HauttypUnbekanntTrockeneHautZweiIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Würdest du dein Hautbild eher rissig und spröde oder
→glänzend oder rosig und zart einschätzen?"
        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

class HauttypUnbekanntNormaleHautZweiHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_normale_haut_antwort_zwei("HauttypUnbekanntNormaleHautZweiIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Würdest du dein Hautbild eher rissig und spröde oder
→glänzend oder rosig und zart einschätzen?"
        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

class HauttypUnbekanntMischhautZweiHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool

```

```

        return ask_utils.
→is_mischhaut_antwort_zwei("HauttypUnbekanntMischhautZweiIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Würdest du dein Hautbild eher rissig und spröde oder
→glänzend oder rosig und zart einschätzen?"
        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

class HauttypUnbekanntFettigeDreiHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_fettige_haut_antwort_drei("HauttypUnbekanntFettigeDreiIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Betrachte mal deine Poren. Sind sie eher groß, klein,
→beides oder hast du keine sichtbaren Poren?"
        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

class HauttypUnbekanntTrockeneDreiHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_trockene_haut_antwort_drei("HauttypUnbekanntTrockeneDreiIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Betrachte mal deine Poren. Sind sie eher groß, klein,
→beides oder hast du keine sichtbaren Poren?"
        return (
            handler_input.response_builder

```

```

        .speak(speak_output)
        # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
        .response
    )

class HauttypUnbekanntNormaleDreiHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_normale_haut_antwort_drei("HauttypUnbekanntNormaleDreiIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Betrachte mal deine Poren. Sind sie eher groß, klein,
→beides oder hast du keine sichtbaren Poren?"
        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

class HauttypUnbekanntMischhautDreiHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_mischhaut_antwort_drei("HauttypUnbekanntMischhautDreiIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Betrachte mal deine Poren. Sind sie eher groß, klein,
→beides oder hast du keine sichtbaren Poren?"
        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

class HauttypUnbekanntFettigeHautVierHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool

```

```

        return ask_utils.
→is_fettige_haut_antwort_vier("HauttypUnbekanntFettigeHautVierIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = ""
        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

class HauttypUnbekanntTrockeneHautVierHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_trockene_haut_antwort_vier("HauttypUnbekanntTrockeneHautVierIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = ""
        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

class HauttypUnbekanntNormaleHautVierHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_normale_haut_antwort_vier("HauttypUnbekanntNormaleHautVierIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = ""
        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

```

```

    )

class HauttypUnbekanntMischhautVierHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_mischhaut_antwort_vier("HauttypUnbekanntMischhautVierIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = ""
        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

class HauttypUnbekanntErgebnisseHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_hauttyp_bekannt_trockene_haut("HauttypUnbekanntErgebnisseIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Okay. Nach deinen Angaben und meinen Analyseergebnissen,
→zufolge hast du eine trockene Haut. Hierfür sind folgende Inhaltsstoffe gut:
→Milchsäure, Urea, Glykolsäure, Ceramide, Sheabutter, Hyaluronsäure, Vaseline
→und Arganöl.Möchtest du eine passende DIY-Maske für deine trockene Haut
→wissen?"
        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

class HauttypUnbekanntErgebnisseHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→is_hauttyp_bekannt_fettige_haut("HauttypUnbekanntErgebnisseIntent")(handler_input)

```

```

def handle(self, handler_input):
    # type: (HandlerInput) -> Response
    speak_output = "Okay. Nach deinen Angaben und meinen Analyseergebnissen,
    ↳zufolge hast du eine fettige Haut. Hierfür sind folgende Inhaltsstoffe gut:
    ↳Salicylsäure, Niacinamid, Hagebutten Kernöl, Mandelsäure, Tonerde,
    ↳Teebaumöl, Benzoylperoxid, Urea und Squalan. Möchtest du eine passende
    ↳DIY-Maske für deine fettige Haut wissen?"
    return (
        handler_input.response_builder
            .speak(speak_output)
            # .ask("add a reprompt if you want to keep the session open for
    ↳the user to respond")
            .response
    )

class HauttypUnbekanntErgebnisseHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
    ↳is_hauttyp_bekannt_normale_haut("HauttypUnbekanntErgebnisseIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Okay. Nach deinen Angaben und meinen Analyseergebnissen,
        ↳zufolge hast du eine normale Haut. Hierfür sind folgende Inhaltsstoffe gut:
        ↳Polyhydroxysäure, weißer Tee, Squalan, Mandelsäure, Panthenol, Arganöl,
        ↳Hyaluronsäure, Glykolsäure und Ceramide. Möchtest du eine passende DIY-Maske
        ↳für deine normale Haut wissen?"
        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
    ↳the user to respond")
                .response
        )

class HauttypUnbekanntErgebnisseHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
    ↳is_hauttyp_bekannt_mischhaut("HauttypUnbekanntErgebnisseIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response

```

```

        speak_output = "Okay. Nach deinen Angaben und meinen Analyseergebnissen,
↳zufolge hast du eine Mischhaut. Hierfür sind folgende Inhaltsstoffe gut:
↳Salicylsäure, Glycerin, Wildrosenöl, Hyaluronsäure, Niacinamid, Jojoba-Öl,
↳Glykolsäure, Aloe Vera und Squalan.Möchtest du eine passende DIY-Maske für
↳deine Mischhaut wissen?"

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
↳the user to respond")
                .response
        )

class MaskeNormalHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
↳is_hauttyp_bekannt_normal("MaskeNormalIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Perfekt! Die passende Maskenrezepte zu deiner normale,
↳Haut lautet:Eine Gurke schälen und pürieren. Anschließend drei Esslöffel
↳gekühlten Joghurt oder Quark unterrühren. Die Gesichtsmaske auf das Gesicht
↳auftragen und für 20 Minuten einwirken lassen. Anschließend mit lauwarmem
↳Wasser sanft, aber gründlich abwaschen.Ich hoffe, ich konnte dir
↳weiterhelfen. "

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
↳the user to respond")
                .response
        )

class MaskeTrockenHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
↳is_hauttyp_bekannt_trocken("MaskeTrockenIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response

```

```

        speak_output = "Perfekt! Die passende Maskenrezepte zu deiner trockene_
↳Haut lautet:Für dieses Rezept brauchst du eine halbe Avocado, einen_
↳Teelöffel frisch gepressten Zitronensaft und ein Eiklar. Vermenge Avocado_
↳mit Zitronensaft. Füge dazu ein Eiklar und rühre es gut zusammen. Gebe die_
↳Mischung auf das gereinigte Gesicht und wasche es nach 20 Minuten mit_
↳lauwarmem Wasser wieder ab. Ich hoffe, ich konnte dir weiterhelfen. "

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for_
↳the user to respond")
                .response
        )

class MaskeMischhautHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
↳is_hauttyp_bekannt_mischhaut("MaskeMischhautIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Perfekt! Die passende Maskenrezepte zu deiner Mischhaut_
↳lautet:Mache aus den zwei Bananen einen Brei. Danach gebe 2 Esslöffel Honig_
↳hinzu und 2 Esslöffel gemahlene Mandeln. Verrühre das Ganze gut. Jetzt_
↳kannst Du die Maske auf dein Gesicht auftragen und für 15 Minuten einwirken_
↳lassen.Ich hoffe, ich konnte dir weiterhelfen."

        handler_input.response_builder
            .speak(speak_output)
            # .ask("add a reprompt if you want to keep the session open for_
↳the user to respond")
            .response
        )

class MaskeFettigHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
↳is_hauttyp_bekannt_fettig("MaskeFettigIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response

```



```

        speak_output = "Perfekt! Die passende Maskenrezepte zu deiner fettigen_
↳Haut lautet:Eine Gurke schälen und pürieren. Anschließend drei Esslöffel_
↳gekühlten Joghurt oder Quark unterrühren. Die Gesichtsmaske auf das Gesicht_
↳auftragen und für 20 Minuten einwirken lassen. Anschließend mit lauwarmem_
↳Wasser sanft, aber gründlich abwaschen.Ich hoffe, ich konnte dir_
↳weiterhelfen. "

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for_
↳the user to respond")
                .response
        )

class NeunzigSekundenLymphdrainageHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
↳neunzig_sekunden_lymphdrainage("NeunzigSekundenLymphdrainageIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response

```

```

speak_output = "90 Sekunden-Lymphdrainage ist eine Art von Massage, die
↳die natürliche Drainage der Lymphe stimuliert und Gifte, wie Bakterien,
↳Viren und Mikroorganismen, aus dem Körper eliminiert. So geht man vor:
↳Erstens: Starte von der Mitte deiner Stirn und benutze hierfür deine Mittel-
↳und Zeigefinger. Drücke hierbei vorsichtig an diesem Punkt. Wiederhole es
↳dreimal. Nun, ziehe langsam deine Finger entlang deiner Augenbrauen bis zu
↳deinen Schläfen und wiederhole dies auch dreimal. Zweitens: Wiederhole die
↳Bewegung, aber diesmal unter deine Augenbrauen. Starte zwischen deine Augen,
↳bei der Nasenrücke, und zieh es wieder Richtung deine Schläfe, wo du dann
↳wieder dreimal mit deinen Fingern Druck ausübst. Diese Kombination von
↳Massagetechniken hilft nicht nur Falten vorzubeugen, sondern es hilft auch
↳gegen chronische Migräne. Drittens: Der nächste Schritt beginnt wieder bei
↳deinem Nasenrücken, wobei du jetzt unter deine Augen arbeitest. Zieh deine
↳Mittel- und Zeigefinger unter deine Augen entlang, bis zum Ende von deinem
↳Auge und übe wieder Druck aus. Zum Schluss verbinde die letzten zwei
↳Schritte: Beginn wieder bei deiner Nasenrücke und kreise mit deine Fingern
↳um deine Augen herum, während du hierbei wieder vorsichtig drückst. Viertens:
↳Führe deine Finger an die Seiten deiner Nase und drücke dreimal. Danach
↳ziehe deine Fingern entlang dem unteren Teil deiner Wangen, bis zur
↳Mundspitze und übe währenddessen Druck aus. Wiederhole diesen Schritt dreimal.
↳Fünfstens: Zum Schluss greife nach deinem Kinn, drücke mit deinen
↳Fingerspitzen dreimal und führe dies entlang der Kinnlinie bis zu deinen
↳Ohren. m ein deutliches und sichtbares Ergebnis zu sehen, müssen diese
↳Schritte zwei bis dreimal in der Woche wiederholt werden. Danke, dass du
↳mitgemacht hast! Bis zum nächsten Session! Ich hoffe du hattest Spaß mit
↳dieser Routine! Willst du zum Hauptmenü zurückkehren? "

```

```

    return (
        handler_input.response_builder
            .speak(speak_output)
            # .ask("add a reprompt if you want to keep the session open for
↳the user to respond")
            .response
    )

```

```

class FaceMessageZuhauseHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
↳face_message_zuhause("FaceMessageZuhauseIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response

```

```

        speak_output = "Diese Routine ist super geeignet wenn es mal schnell␣
        ↳gehen muss. Lass uns beginnen!Erstens: Reinige deine Hände und dein Gesicht.
        ↳Zweitens: reibe ein bisschen Öl oder Essenz in deinen Händen und trage es␣
        ↳vorsichtig auf deinem Gesicht auf.Drittens: Massiere den Lymph- Bereich␣
        ↳unter deinen Ohren und entlang der Seiten von deinem Hals und Nacken.␣
        ↳Viertens: Knete vorsichtig deine Wangen und die Seiten deines Gesichts.
        ↳Fünftens: Führe deine Finger entlang deiner Stirn, drück deine Fingerspitzen␣
        ↳über deine Augenbrauen und leite sie langsam nach unten Richtung deiner␣
        ↳Schläfen.Sechstens: führe dann deine Finger unter deine Augen und leite sie␣
        ↳langsam, mit einem bisschen Druck nach oben, wieder Richtung deiner Schläfen.
        ↳Zum Schluss verwende leichte, vertikale bewegugen und massiere deinen␣
        ↳Nacken- und Dekollettebereich. Danke, dass du mitgemacht hast! Ich hoffe du␣
        ↳hattest Spaß mit dieser Routine! Willst du zum Hauptmenü zurückkehren? "

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for␣
        ↳the user to respond")
                .response
        )

class KoreanischeRoutineHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
        ↳koreanische_routine("KoreanischeRoutineIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response

```

```

        speak_output = "Schön das du dich dafür entschieden hast! Für diese
        ↳Routine benötigst du: Reinigungsöl, Cleanser, Peeling, Toner, Wattepads,
        ↳Essenz, Augencreme, Serum, Gesichtsmaske, Hautcreme und Sonnencreme. Als
        ↳erstes reinige dein Gesicht mit einem Reinigungsöl um dein Makeup restlos zu
        ↳entfernen. Benutze einen Cleanser mit lauwarmen Wasser, um deine Poren frei
        ↳von Verstopfungen zu kriegen. Diesen Schritt kannst du zweimal wiederholen,
        ↳um das beste Ergebnis zu erzeugen. Als dritten Schritt ist Peeling dran: Ein
        ↳bis zwei Mal pro Woche sollte deine Haut von abgestorbenen Hautschüppchen
        ↳befreit werden. Tropfe Toner auf einem Wattepad und wische deine Haut damit,
        ↳um den pH-Wert wieder herzustellen. Als fünftens kommt Essenz dran. Dabei
        ↳werden die Inhaltsstoffe in ihre einzelnen Aminosäuren, Vitamine und
        ↳ähnliche Bestandteile aufgespalten und können so besonders intensiv in die
        ↳Haut eindringen. Um Augenringe und später Falten zu vermeiden, nutze deine
        ↳Augencreme um deine Augen und deine Augenbrauen. Sechstens: Das Serum ist ein
        ↳wichtiger Bestandteil. Trage überall dein Serum deines Hauttyps auf und lass
        ↳diese einwirken. Siebtens: Für ein extra Spa-Erlebnis kannst du eine
        ↳Gesichtsmaske nutzen. Zum Schluss pflege deine Haut mit einer Creme deiner
        ↳Wahl die zu deinem Hauttyp passt. Vergiss nicht einen UV-Schutz zu
        ↳verwenden! Danke, dass du mitgemacht hast! Ich hoffe du hattest Spaß mit
        ↳dieser Routine! Willst du zum Hauptmenü zurückkehren? "

```

```

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
        ↳the user to respond")
                .response
        )

```

```

class GuaShaHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.gua_sha("GuaShaIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response

```

```

        speak_output = "Willkommen bei der Gua-Sha-Massage! Diese Technik hilft,
        ↳ dir dein Gesicht zu formen und Falten vorzubeugen. Hast du keinen,
        ↳ Gua-Sha-Stein zur Hand? Dann benutze einfach deine Finger oder einen Löffel,
        ↳ dafür. Lass uns anfangen! Erstens: verwende ein sauberes Gua-Sha-Stein auf,
        ↳ deine vorher gereinigte Haut. Zweitens: Der Stein sollte auf deiner Haut,
        ↳ gleiten, falls nicht, trage ein wenig Öl oder Creme auf deinem Gesicht und,
        ↳ Hals auf. Die Grundregel bei der Gua-Sha Technik ist, dass du den Stein,
        ↳ immer von innen nach außen streifst, um den Lymphfluss anzuregen. Nur am,
        ↳ Hals kannst du von unten nach oben massieren. Drittens: Streife den,
        ↳ Gua-Sha-Stein leicht angeschrägt, über die Haut. Beginne von deinem Kinn aus,
        ↳ zieh den Stein zu deinen Ohren, so entspannst du deine Kiefer. Viertens:,
        ↳ Wiederhole das Ganze von deiner Nase bis zum Haaransatz. Fünftens: Streife,
        ↳ von der Nasenwurzel zu den Schläfen. Reduziere dabei den Druck, um die deine,
        ↳ Haut nicht zu strapazieren. Sechstens: Zum Schluss streife den Stein von der,
        ↳ Mitte deiner Stirn über deine Schläfe bis hin zum Haaransatz. Bei,
        ↳ Kopfschmerzen tut es gut, den Stein von der Stirn aus über die Kopfhaut zu,
        ↳ ziehen. Um eine deutliche und sichtbare Ergebnis zu sehen, müssen diese,
        ↳ Schritte zwei bis dreimal in der Woche wiederholt werden . Danke, dass du,
        ↳ mitgemacht hast! Bis zum nächsten Session! Ich hoffe du hattest Spaß mit,
        ↳ dieser Routine! Willst du zum Hauptmenü zurückkehren? "

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for,
        ↳ the user to respond")
                .response
        )

class TippTrockeneLippenHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
        ↳ tipp_trockene_lippen("TippTrockeneLippenIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Hat man gerade kein Pflegestift oder Vaseline zur Hand,
        ↳ so gibt es auch andere schnelle Methoden, um geschmeidige Lippen zu bekommen.
        ↳ Als Soforthilfe für zu Hause eignen sich Quark oder Honig super. Trägst du,
        ↳ sie auf deine trockenen Lippen auf, machen sie die Lippen zart und,
        ↳ geschmeidig. Gegen spröde Lippen hilft auch ein Peeling mit Öl und Zucker.
        ↳ Dafür kannst du einfach Olivenöl mit braunem Zucker vermengen und sanft in,
        ↳ die Lippen massieren. Kleiner Tipp: in der Nacht regeneriert der Körper,
        ↳ weswegen das die beste Zeit ist, um die Lippen zu pflegen. Ich hoffe ich,
        ↳ konnte dir weiterhelfen, gibt es noch etwas, was dir Schwierigkeiten,
        ↳ bereitet?"

        return (

```

```

        handler_input.response_builder
            .speak(speak_output)
            # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
            .response
    )

class TippGesichtshaareHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→tipp_gesichtshaare("TippGesichtshaareIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Möchtest du es eher weniger schmerzhaft haben, so
→kannst du deine Gesichtshaare mit einem Rasierer rasieren oder eine
→Enthaarungscreme anwenden. Du musst aber bedenken dass die Härchen je nach
→Stärke des Haarwachstums schon nach paar Tagen wieder rauswachsen können.
→Für einen langfristigen Effekt sollte die Haarwurzel entfernt werden und das
→kann schmerzhaft sein. Als gängige Methoden hierfür zählen das Wachsen, die
→Fadentechnik und auch immer mehr im Trend die Laser-Haarentfernung. Ich hoffe
→ich konnte dir weiterhelfen, gibt es noch etwas, was dir Schwierigkeiten
→bereitet?"
        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
→the user to respond")
                .response
        )

class TippEingewachseneHaareHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→tipp_eingewachsene_haare("TippEingewachseneHaareIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response

```

```

        speak_output = "Eingewachsene Haar solltest du mit einer sterilen Nadel,
↳ an die Oberfläche bringen und sie mit einer sauberen Pinzette entfernen. Um,
↳ eingewachsene Haare vorzubeugen kann dir das regelmäßige Peelen helfen.
↳ Dadurch werden die Poren nicht verstopft und Entzündungen vorgebeugt. Das,
↳ Rasieren stresst die Haut und Haarwurzeln und führt deshalb besonders zu,
↳ eingewachsenen Haaren. Solltest du dich weiterhin rasieren, solltest du,
↳ darauf achten, die Haare in der Wuchsrichtung statt gegen die Wuchsrichtung,
↳ zu entfernen. Ich hoffe ich konnte dir weiterhelfen, gibt es noch etwas, was,
↳ dir Schwierigkeiten bereitet?"

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for,
↳ the user to respond")
                .response
        )

```

```

class TippHerpesHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.tipp_herpes("TippHerpesIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Eine zinkhaltige Salbe kann im Vorfeld der,
↳ Bläschenbildung die Infektion abmildern und im besten Fall den Herpes ganz,
↳ verhindern. Zink trocknet auch die betroffenen Stellen aus. Eine weitere,
↳ Möglichkeit ist Honig. Honig tötet Bakterien und Viren ab und kann so eine,
↳ Verschlimmerung vermeiden. Ich hoffe ich konnte dir weiterhelfen, gibt es,
↳ noch etwas, was dir Schwierigkeiten bereitet? "

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for,
↳ the user to respond")
                .response
        )

```

```

class TippWundeHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.tipp_wunde("TippWundeIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response

```

```

        speak_output = "Hast du eine Wunde im Gesicht, solltest du sie zuerst_
→einmal reinigen und dann Zinksalbe auftragen. Ein anderes effektives Mittel_
→ist Teebaumöl, dieser wirkt für deine Wunde entzündungshemmend und sollte_
→dünn aufgetragen werden. Hast du keines dieser Stoffe zuhause, so kannst du_
→auch Honig verwenden. Bei großen und tiefen Wunden solltest du das am besten_
→bei einem Arzt überprüfen lassen, die Infektionsgefahr ist hier höher und es_
→könnte eine professionelle Therapie nötig sein. Ich hoffe ich konnte dir_
→weiterhelfen, gibt es noch etwas, was dir Schwierigkeiten bereitet?"

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for_
→the user to respond")
                .response
        )

class TippGerstenkornHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
→tipp_gersternkorn("TippGerstenkornIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Ein beliebtes Hausmittel gegen Gerstenkorn sind feuchte_
→Kompressen. Sie sollen auf das entzündete Auge aufgelegt oder zum Abtupfen_
→des Auges verwendet werden. Zur Herstellung nimmt man zum Beispiel_
→Kamillentee aufgrund seiner leicht entzündungshemmenden Eigenschaften. Durch_
→Wärme öffnet sich das Gerstenkorn schneller und der Eiter kann leichter_
→abfließen. Damit sich die Infektion nicht ausbreitet, kann der Augenarzt_
→antibiotikahaltige Salben oder Augentropfen verschreiben. Ich hoffe ich_
→konnte dir weiterhelfen, gibt es noch etwas, was dir Schwierigkeiten_
→bereitet?"

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for_
→the user to respond")
                .response
        )

class TippPickelHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.tipp_pickel("TippPickelIntent")(handler_input)

    def handle(self, handler_input):

```



```

        # type: (HandlerInput) -> Response
        speak_output = "Für die Pickelbekämpfung kannst du entzündungshemmende,
        ↳ Salben oder Creme auftragen, dafür eignen sich Zinksalbe oder Aloe Vera sehr
        ↳ gut. Weiter kannst du den Pickel auch austrocknen lassen, mit beispielsweise
        ↳ Rosenwasser oder Teebaumöl. Wichtig ist, den Pickel am besten nicht anfassen.
        ↳ Kannst du nicht widerstehen und möchtest ihn ausdrücken, so wasche deine
        ↳ Hände bitte davor, um weitere Infektionen zu vermeiden. Ich hoffe ich konnte
        ↳ dir weiterhelfen, gibt es noch etwas, was dir Schwierigkeiten bereitet?"

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
        ↳ the user to respond")
                .response
        )

class TippAugenringeHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.tipp_augenringe("TippAugenringeIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Der Klassiker für Augenringe ist die Gurke, davon,
        ↳ schneidest du dir am besten zwei Scheiben ab und legst sie für 10 Minuten
        ↳ auf deine geschlossenen Augen. Alternativ funktionieren auch Schwarz- oder
        ↳ Grüntee. Hierfür legst du zwei Teebeutel kurz ins warme Wasser, lässt sie
        ↳ ziehen und legst sie dann für 10 Minuten auf die Augen. Genug Schlaf und
        ↳ eine gesunde Ernährung essenziell, achte auch darauf! Ich hoffe ich konnte
        ↳ dir weiterhelfen, gibt es noch etwas, was dir Schwierigkeiten bereitet?"

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
        ↳ the user to respond")
                .response
        )

class TippAusschlagHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.tipp_ausschlag("TippAusschlagIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response

```

```

        speak_output = "Bei einem Ausschlag ist es wichtig, diesen zu kühlen,
↳beispielsweise mit Aloe Vera. Falls dein Ausschlag zudem auch jucken sollte,
↳empfehle ich dir Salben und Cremes mit dem Stoff Antihistaminika. Tretet
↳dein Ausschlag öfter auf, kann dies auch allergisch bedingt sein, da würde
↳ich dir empfehlen, dass beim Dermatologen abchecken zu lassen.Ich hoffe ich
↳konnte dir weiterhelfen, gibt es noch etwas, was dir Schwierigkeiten
↳bereitet?"

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
↳the user to respond")
                .response
        )

class TippVerbrennungHandler(AbstractRequestHandler):
    def can_handle(self, handler_input):
        # type: (HandlerInput) -> bool
        return ask_utils.
↳tipp_verbrennung("TippVerbrennungIntent")(handler_input)

    def handle(self, handler_input):
        # type: (HandlerInput) -> Response
        speak_output = "Zuerst einmal kannst du den Schmerz der Brandverletzung
↳lindern und Hautschäden vorbeugen, indem du den betroffenen Bereich mit
↳Wasser kühlst. Ist die Verbrennung stark und tief, solltest du diese mit
↳einem Verband abdecken und im besten Falle auch einen Arzt das abchecken
↳lassen. Was du auch jederzeit bei jeglicher Verbrennung benutzen kannst,
↳sind Wund- und Heilsalben. Feuchtigkeitsspendende Produkte, unterstützen die
↳Heilung und verhindert die Narben.Ich hoffe ich konnte dir weiterhelfen,
↳gibt es noch etwas, was dir Schwierigkeiten bereitet?"

        return (
            handler_input.response_builder
                .speak(speak_output)
                # .ask("add a reprompt if you want to keep the session open for
↳the user to respond")
                .response
        )

sb = SkillBuilder()

sb.add_request_handler(LaunchRequestHandler())
sb.add_request_handler>HelloWorldIntentHandler()
sb.add_request_handler(HelpIntentHandler())
sb.add_request_handler(CancelOrStopIntentHandler())

```

```

sb.add_request_handler(SessionEndedRequestHandler())
sb.add_request_handler(HauttypBekanntHandler))
sb.add_request_handler(HauttypBekanntNormaleHautHandler))
sb.add_request_handler(HauttypBekanntFettigeHautHandler))
sb.add_request_handler(HauttypBekanntTrockeneHautHandler))
sb.add_request_handler(HauttypBekanntMischhautHandler))
sb.add_request_handler(HauttypUnbekanntHandler))
sb.add_request_handler(HauttypUnbekanntFettigeHautHandler))
sb.add_request_handler(HauttypUnbekanntTrockeneHautHandler))
sb.add_request_handler(HauttypUnbekanntNormaleHautHandler))
sb.add_request_handler(HauttypUnbekanntMischhautHandler))
sb.add_request_handler(HauttypUnbekanntFettigeHautZweiHandler))
sb.add_request_handler(HauttypUnbekanntTrockeneHautZweiHandler))
sb.add_request_handler(HauttypUnbekanntNormaleHautZweiHandler))
sb.add_request_handler(HauttypUnbekanntMischhautZweiHandler))
sb.add_request_handler(HauttypUnbekanntFettigeDreiHandler))
sb.add_request_handler(HauttypUnbekanntTrockeneDreiHandler))
sb.add_request_handler(HauttypUnbekanntNormaleDreiHandler))
sb.add_request_handler(HauttypUnbekanntMischhautDreiHandler))
sb.add_request_handler(HauttypUnbekanntFettigeHautVierHandler))
sb.add_request_handler(HauttypUnbekanntTrockeneHautVierHandler))
sb.add_request_handler(HauttypUnbekanntNormaleHautVierHandler))
sb.add_request_handler(HauttypUnbekanntMischhautVierHandler))
sb.add_request_handler(HauttypUnbekanntErgebnisse))
sb.add_request_handler(MaskeNormal))
sb.add_request_handler(MaskeTrocken))
sb.add_request_handler(MaskeMischhaut))
sb.add_request_handler(MaskeFettig))
sb.add_request_handler(IntentReflectorHandler())
sb.add_request_handler(NeunzigSekundenLymphdrainageHandler))
sb.add_request_handler(FaceMassageZuhauseHandler))
sb.add_request_handler(KoreanischeRoutineHandler))
sb.add_request_handler(GuaShaHandler))
sb.add_request_handler(TippTrockeneLippenHandler))
sb.add_request_handler(TippGesichtshaareHandler))
sb.add_request_handler(TippEingewachseneHaareHandler))
sb.add_request_handler(TippHerpesHandler))
sb.add_request_handler(TippWundeHandler))
sb.add_request_handler(TippGerstenkornHandler))
sb.add_request_handler(TippPickelHandler))
sb.add_request_handler(TippAugenringeHandler))
sb.add_request_handler(TippAusschlagHandler))
sb.add_request_handler(TippVerbrennungHandler))

# make sure IntentReflectorHandler is last so it doesn't override your custom
↳ intent handlers

```

```
sb.add_exception_handler(CatchAllExceptionHandler())

lambda_handler = sb.lambda_handler()
```

```
-----
ModuleNotFoundError                                Traceback (most recent call last)
/tmp/ipykernel_175/714281275.py in <module>
      6 # This sample is built using the handler classes approach in skill_
      ↪builder.
      7 import logging
----> 8 import ask_sdk_core.utils as ask_utils
      9
     10 from ask_sdk_core.skill_builder import SkillBuilder

ModuleNotFoundError: No module named 'ask_sdk_core'
```

```
[ ]:
```