

# S07: Skill selbst hosten

Während der Entwicklung kann es einfacher sein, das Backend für den Skill auf dem eigenen Rechner laufen zu lassen. (Wo es dann korrigiert und neugestartet werden kann.) Dazu müssen zwei Sachen eingerichtet werden:

- Der Code muß statt über Lambda angesprochen zu werden, über ein Python-Webservice-Framework angesprochen werden.
- Der auf dem eigenen Rechner laufende Webservice muß für das Alexa Skill Kit als Endpoint erreichbar gemacht werden.

## Flask als Webservice-Framework

Es muss `flask-ask` installiert werden, über: `pip install flask-ask-sdk`

Der Code muß dann angepasst werden, siehe `flaskask.py`. (Das ist meine angepasste Version des [python-helloworld-classes-Skills](#), entsprechend findet sich dort auch das Interaktionsmodell.)

Bei Ausführung wird dann auf dem eigenen Rechner (`localhost`) auf Port 5000 der Webservice ausgeführt.

## Eigenen Rechner erreichbar machen

Damit ASK auf diesen Webservice zugreifen kann, muss dieser Port nach außen weitergeleitet werden. Dazu benutzen wir das Programm `ngrok`. Dieses müssen Sie auf Ihrem Rechner noch [installieren](#).

Wenn der Webservice wie im vorigen Abschnitt beschrieben läuft, kann mit `ngrok http 5000` ein Tunnel eingerichtet werden, der den eigenen Port 5000 an einen von `ngrok` betriebenen Rechner weiterleitet.

Nach Ausführung dieses Kommandos werden Ihnen die URLs angezeigt, unter denen ihr Rechner (nur dieser Port, keine Angst!) erreichbar ist. Kopieren Sie die `https` URL und fügen Sie sie als Endpoint im Skillbuilder ein. Wählen Sie dort auch "wildcard certificate" aus.

Jetzt sollte das auf ihrem Rechner laufende Programm von Alexa aus erreichbar sein. Wenn sie an dem Programm etwas ändern wollen, können Sie es einfach unterbrechen und dann wieder neustarten. Der mit `ngrok` eingerichtete Tunnel muß (und sollte) dabei nicht unterbrochen werden. (Denn sonst bekommen Sie beim Neustart eine neue URL zugewiesen, die Sie dann wieder als Endpoint eintragen müssten.)