



openHPI Workshop Embedded Smart Home 2017

Einführung Webframework Flask

Martin Malchow & Matthias Bauer

Client-Server-Prinzip

Client stellt Anfrage an Server (Request),
Server beantwortet Anfrage (Response)

Konkret:

- Webserver auf dem Raspberry Pi
- Client (zunächst) auch → schickt Anfragen an Server
- Server generiert Webseite
- Server kann Code (Python) ausführen
- Server schickt Client die Webseite
- Client stellt Webseite im Browser dar

**Embedded Smart
Home 2017**

Martin Malchow
Matthias Bauer

Flask – Ein (Micro-)Webframework für Python

Kümmert sich um die grundlegende Serverfunktionalität

- Öffnet einen Socket
- wartet auf TCP-Verbindungen
- nimmt HTTP-Requests an
- analysiert die Anfrage und ruft von uns definierte Funktionen auf, die Antwort-HTML generieren
- schickt eine entsprechende HTTP-Antwort

Aufgabe: Funktionen, die abhängig vom jeweiligen URL

- auf dem Server Code ausführen und
- HTML zurückgeben

**Embedded Smart
Home 2017**

Martin Malchow
Matthias Bauer

Template Engine

- Werden bei Darstellung unterstützt durch Template Engine
- Gerüst bauen, in das wir bestimmte Werte einsetzen
- Gerüst stellt grundlegende Kachelfunktionalität zur Verfügung
- Python-Code generiert Farben und Texte und lässt diese dann von der Template Engine einsetzen
- Grundlegendes Template sowie viele verwendbare Funktionalitäten durch Bootstrap genommen (z.B. Carousel, Row/Column-Anordnung)

**Embedded Smart
Home 2017**

Martin Malchow
Matthias Bauer

Einrichtung, weitere Informationen

Vorbereiteten Code auf Github runterladen

- <https://github.com/openHPI/Embedded-Smart-Home-2017>

Flask Homepage

- <http://flask.pocoo.org>

**Embedded Smart
Home 2017**

Martin Malchow
Matthias Bauer

Virtual Environment

1. installieren: `sudo pip3 install virtualenv`
2. erstellen: `virtualenv venv`
3. aktivieren: `source venv/bin/activate`
deaktivieren: `deactivate`

Flask und die -Bootstrap-Erweiterung installieren

- `sudo pip3 install flask flask-bootstrap.`

Hauptprogramm zu Environment-Variable exportieren

- `export FLASK_APP=smarthome.py`

**Embedded Smart
Home 2017**

Martin Malchow
Matthias Bauer

Server starten

Flask starten

- `python3 -m flask run`
- Auf Konsole werden Logininformationen ausgegeben
- im Browser `http://localhost:5000` aufrufen
- `[ctrl] + [C]` zum Beenden des Servers

**Embedded Smart
Home 2017**

Martin Malchow
Matthias Bauer