



Universität Regensburg

**Philosophische Fakultät III**  
**Sprach- , Literatur- und Kulturwissenschaften**  
**Institut für Information und Medien, Sprache und Kultur (I:IMSK)**  
**Lehrstuhl für Medieninformatik**

---

Seminar KI für Serious Games  
Modul: INF-M07.2  
Leitung: Professor Bernd Ludwig  
Abgabedatum: 31.03.2018

## **„How To“ – Handbuch**

David Halbhuber 1744590  
Jakob Fehle 1770881

## **Inhaltsverzeichnis:**

<b>1</b>	<b>Motivation .....</b>	<b>3</b>
<b>2</b>	<b>Verwendung Schnittstelle zur Nährstoffberechnung (API).....</b>	<b>3</b>
2.1	Installation Flask .....	3
<b>3</b>	<b>Verwendung des Frontend/Datenbanksuche .....</b>	<b>4</b>

## 1 Motivation

Das vorliegende Dokument dient als Handbuch für die im Rahmen des Seminars „KI for serious games“ entstandene Software. Diese Anweisung wurde gefertigt, um einen schnellen Start in die von uns entwickelten Anwendungen zu ermöglichen. Zudem soll dieses Dokument erläutern wie die entwickelten Dienste zu einem späteren Zeitpunkt erneut gestartet werden können.

## 2 Verwendung Schnittstelle zur Nährstoffberechnung (API)

Diese entwickelte Schnittstelle soll anderen Benutzern und Entwicklern gestatten Nährwerte von neuen Gerichten und Rezepten auszuwerten, ohne die Rezepte vorher in ein .raw Format umzuwandeln. Dabei dient die Python Bibliothek Flask als Webframework und build-in Webserver. Da zum Zeitpunkt der Abgabe dieser Projektarbeit Flask nicht auf dem Universitätsserver bereitgestellt werden konnte, wird im Folgenden kurz dessen Installation erläutert und anschließend geschildert wie die Webapplikation schlussendlich gestartet werden kann.

### 2.1 Installation Flask

Flask kann mit Hilfe von Python Paketmanagers „pip“ installiert werden, dazu genügt folgender Linux Befehl:

- `sudo python pip install flask`

Nach der Installation von Flask kann der Webserver durch folgende Linux Befehl gestartet werden:

- Wechsel in korrektes Verzeichnis
  - `Cd /home/kochbar/server/server/src`
    - `python api.py`

Die gestartete Anwendung horcht nun auf Serverport 5000, konkret erreichbar also unter <http://132.199.138.70:5000> . Für den Fall, dass der Dienst ausfällt, muss Flask nicht neuinstalliert werden - es kann direkt mit dem zweiten Schritt begonnen und der Webserver gestartet werden. Sobald der Dienst unter oben genannter Adresse erreichbar ist,

erlaubt die API REST Anfragen (POST :5000/get\_nutritions) mit einem JSON-String als Übergabe Parameter, der aus einem Identifier-String und einem Zutaten-String besteht. Eine beispielhafte JSON-Datei kann im verwendeten Github-Repository<sup>1</sup> gefunden werden. Die API liefert dann eine Breite Auswahl an herkömmlichen Nährwerten zurück. Prinzipiell kann Flask, wie jede andere Python Bibliothek, auch in einer virtuellen Umgebung installiert und gestartet werden, dies macht vor allem dann Sinn, wenn Flask nicht global zur Verfügung gestellt werden soll.

### 3 Verwendung des Frontend/Datenbanksuche

Die entworfene Datenbanksuche, das Frontend, ist am Universitätsserver auf Port 9009 erreichbar, die Adresse ist folglich: <http://132.199.138.79:9009> . Zur Bereitstellung verwenden wir hierfür **node.js**. Zum aktuellen Zeitpunkt läuft der Webserver und ist aus dem Uni-Netz erreichbar. Für den Fall, dass der Dienst nicht mehr erreichbar ist, wie im Falle eines Serverneustarts, kann die Webanwendung mit folgendem Linux Befehl erneut gestartet werden.

- Wechsel in das korrekte Verzeichnis
  - CD /home/kochbar/frontend\_final/dbAPI
- Starten des node.js Servers
  - screen
    - nodejs server.js
      - STRG+A
        - D
- Der Webserver läuft nun im Hintergrund und sollte über den Port 9009 erreichbar sein.

---

<sup>1</sup> Beispiel Json-Datei: <https://github.com/JakobFehle/KI4SG/blob/master/json/testparse.json>

Für die Darstellung der Rezepte, die Speicherung der Nutzereingabe (also die Auswahl der Gerichte), sowie für die Berechnungen des Nährwertbedarfs und des Nährwertverbrauchs, verwendet die Anwendung drei verschiedene SQL-Tabellen.

- `kochbar.referenzenuts`
  - Speichert die empfohlen Tagesmengen der Nährwerte in Abhängigkeit der Nutzergruppe (Alter, Geschlecht) ab
- `kochbar.users`
  - Ordnet Nutzer zu verzehrten Gerichten zu
- `kochbar.recipenuts`
  - Diese Tabelle speichert die von uns berechneten Nährwerte der ca. 300000 Rezepte des ursprünglichen kochbar-Datensatzes. Diese Tabelle wird auch für die Darstellung der Nährwerte in der Suchergebnisanzeige verwendet, sowie zu schlussendlichen Nährwertberechnung für den individuellen Benutzer.