



# Ein REST API Server in Python

## 1 Vorbereitung

Starten Sie eine Shell und wechseln Sie in das Verzeichnis Desktop/Kursfiles/RESTAPIServer. Die hier zur Verfügung gestellte Datei RESTAPI\_Server.py stellt einen REST API Server da. Dazu wird die Python-Bibliothek flask eingesetzt. Geben Sie den Code mit cat auf der Shell aus und versuchen Sie ihn zu verstehen. Welche Grundrechenarten beherrscht das API aktuell?  
Der Server wird aktiviert mit dem Befehl:

```
python3 RESTAPI_Server.py
```

Mit Strg + c wird er wieder deaktiviert.

## 2 Übungen

### 2.1 Übung 1

Starten Sie eine weitere Shell und wechseln Sie in das Verzeichnis Desktop/Kursfiles/RESTAPIServer, um damit den Client zu bedienen. Verwenden Sie das Skript RESTAPI\_Client.py, um API-Calls an den Server zu richten. Geben Sie zunächst mit cat den Code auf der Kommandozeile aus und analysieren Sie ihn. Welche Rechenarten werden aktuell durch den Client unterstützt?  
Rufen Sie nun den Client auf. Die zwei Zahlen und der Operator werden als Argumente des Skripts übergeben. z.B.

```
python3 RESTAPI_Client.py 4 '+' 5
```

Der Operator muss hier in Single Quotes `'...'` eingefasst werden, um Ambiguitäten mit Sonderzeichen der Linux-Shell zu vermeiden.

### 2.2 Übung 2

Erweitern Sie den Server RESTAPI\_Server.py, so dass auch die Multiplikation und die Division per API-Call möglich sind. Öffnen Sie dazu die Datei mit Pluma. Die Grundstruktur des Codes lässt sich von den beiden Ressourcen (Addition und Subtraktion) übernehmen. Elemente wie URL und Funktionsname sind anzupassen, damit sie für die anzulegenden Ressourcen eindeutig sind. (Der Server startet automatisch neu, wenn Änderungen an der Datei vorgenommen werden.)

Verfahren Sie ebenso mit dem Client und testen Sie dann die neuen Grundrechenarten.

## 2.2 Übung 3 (optional)

Ändern sie im Code für Server und Client die Methode auf GET (oder DELETE) und verifizieren Sie dann die Funktion. Welche Erkenntnis bzgl. REST API können sie daraus ableiten?