

RESTful Interfaces und Templates mit Flask

Einleitung

Im Folgenden soll eine kleine Flask-Anwendung gebaut werden die ein RESTful Interface implementiert. Das Thema dreht sich um den Stundenplan der FH Hof, den wir mit einem Python-Skript von der Website geparkt haben, und als JSON im Ordner `data/` bereitstellen.

Für diese Übung müsst ihr lediglich `app.py` anpassen. Momentan besteht diese aus dem Flask HelloWorld.

Ausführen könnt ihr eure Anwendung indem ihr in das Directory `practice` navigiert, und dort `app.py` ausführt. Wenn alles gut ging, dann könnt ihr unter <http://localhost:5000> in euren Webbrowser eine Begrüßungsmassage sehen.

Testen könnt ihr die Aufgaben **a)**, **b)** und **c)** indem ihr `test.py` ausführt:

```
$ python test.py a b c
** Querying: http://localhost:5000/api/count/all
** Querying: http://localhost:5000/api/count/Inf
** Querying: http://localhost:5000/api/list
** Querying: http://localhost:5000/api
...

-----

Ran 4 tests in 0.016s

OK
```

Aufgabe **e)** können wir nach Termin gemeinsam testen.

Die Musterlösung stellen wir gegen Ende der Stunde dann auf Dropbox.

Aufgaben

a)

Erweitere `app.py` so dass eine Abfrage von <http://localhost:5000/api/Inf/5> unseren aktuellen Stundenplan liefert. Der Studiengang `Inf` und das Semester soll dabei entsprechend durch andere Studiengänge wie `BW` ersetzt werden können.

Beispiel Antwort:

```
$ curl localhost:5000/api/Inf/5
{
  "Dienstag": [{
    "room": "FB102",
    "prof": "Prof. Dr. J\u00f6rgen Heym",
    "time": "11:30-13:00",
    "name": "RoutingSwitchingTroubleshooting 1",
    "note": "(Inf+MI+WI5)",
    "type": "FWM:1"
  },
  ...
  ]
}
```

Für das Abfragen eines Python Dictionaries mit dem Stundenplan könnt ihr die Funktion `load(studiengang, semester)` nutzen. Für das Umwandeln in valides JSON könnt ihr die eingebaute Funktion `json.dumps(python_object)` nutzen.

Wird ein Stundenplan von einem nichtexistierenden Kurs geholt, so soll ein leeres Dictionary zurückgegeben werden. `{}`. Für nichtexistierende Kurse löst die `load`-Funktion eine `NoSuchCourse` Exception aus.

b)

http://localhost:5000/api/list_courses soll eine sortierte Liste aller Studiengänge wiedergeben.

Beispiel:

```
$ curl localhost:5000/api/list_courses
["BBB", "BW", "GP", "IM", "Inf", ..., "Wing MT", "Wing WT"]
```

Zum Abfragen der unsortierten Kursliste könnt ihr die Funktion `list_courses()` verwenden.

c)

<http://localhost:5000/api/count/studiengang> soll die Anzahl der aktuell studierenden Semester für einen Studiengang ausgeben.

Für `Inf` wäre das beispielsweise **3** (`== len('Inf1', 'Inf3', 'Inf5')`).

Für den speziellen Studiengang `all` sollen alle Studiengänge gezählt werden.

Beispiel:

```
$ curl localhost:5000/api/count/Inf
3
$ curl localhost:5000/api/count/all
84
```

Zum Abfragen der Anzahl könnt ihr die bereitgestellte Funktion `count(studiengang)` nutzen. Das Argument `studiengang` ist optional. Lässt man es weg werden alle Studiengänge gezählt.

d)

Erweitere die Flask-Anwendung um eine weitere URL:

<http://localhost:5000/view/studiengang/semester>.

Diese soll bei Aufruf von beispielsweise `view/Inf/5` unseren Stundenplan rendern.

Nutze dazu die Methode `render_template()` und schreibe ein Jinja2 Template dass fähig ist den Stundenplan als HTML zu rendern.

In `templates/simple_table.html` ist eine leichte Hilfestellung bzgl. HTML Tables. Ein brauchbares CSS ist bereits eingebaut.

Ihr müsst euch nicht ans Template halten, ihr könnt es beliebig verändern.

Beispiel Bild:

Montag		
Raum	Zeit	Name
Ex_Virtuell	04:00-05:30	Gender und Diversity
FB010	09:45-11:15	ObjectiveC und iOSProgrammierung
FB102	11:30-13:00	TCP/IP Routing I
FA014	11:30-13:00	Signalverarbeitung
FA012	14:00-15:30	Programming Windows Phone
FA013	17:30-19:00	Wissens, Innovations und Technologiemanagement
FD177	17:30-19:00	Wiederverwendungsbasierte Entwicklung von Systemen
FB001/002	18:00-19:30	Unternehmen in der Region HochFranken
FB001/002	18:00-19:30	Unternehmen in der Region HochFranken
Ex_Extern	18:00-19:30	Unternehmen in der Region HochFranken
Dienstag		
Raum	Zeit	Name
FB010	08:00-09:30	ObjectiveC und iOSProgrammierung
FB102	11:30-13:00	Routing/Switching/Troubleshooting 1
FB114	11:30-13:00	Oracle 11g Release 2 Administration Workshop I
FA017	14:00-15:30	Wiederverwendungsbasierte Entwicklung von Systemen
FA013	17:30-19:00	Meteorologie nicht nur für Privatpiloten
FB023	19:15-20:45	Grundlagen der Medizin und der Medizintechnik für Informatiker und Ingenieure

e)

Bitte backt uns eine Python-Torte:

<http://www.cakefriday.de/2012/03/python-cake.html>

