

---

# A07: Python Webframeworks

## CherryPy - CRUD

---

SEW  
5BHIT 2017/18

Marvin Ertl, Lukas Zuba

Note:  
Betreuer:

Version 1.0  
Begonnen am 22. Dezember 2017  
Beendet am 11. Januar 2018

# Inhaltsverzeichnis

<b>1</b>	<b>Aufgabenstellung</b>	<b>1</b>
<b>2</b>	<b>Umsetzung</b>	<b>2</b>
2.1	CherryPy . . . . .	2
2.2	Tabelle . . . . .	3
2.3	Aufzählung . . . . .	3
2.4	Code . . . . .	3

# 1 Aufgabenstellung

Suche dir mit einem/einer Partner/in ein Python Webframework aus und präsentiere deine Lösung!  
Grundanforderungen (70%):

- Installiere und konfiguriere eines der präsentierten Frameworks
- Überlege dir einen sinnvollen Anwendungsfall für das Framework und erstelle eine passende Datenbank dazu
- Erstelle eine simple Seite, welche Datensätze aus einer Datenbank anzeigt
- Protokolliere deine Vorgehensweise, aufgetretene Probleme etc.

Erweiterungen (30%):

- Verwende ein ansprechendes Design
- Die Seite erlaubt zusätzlich:
  - Das Bearbeiten von Datensätzen
  - Das Löschen von Datensätzen
  - Das Erstellen von neuen Datensätzen

Abgabe: Protokoll inkl. Arbeitsaufwand, Screenshots und Beschreibungen

## 2 Umsetzung

Die Übung ist in 2 Bereiche aufzuteilen, den Server Teil, dieser kümmert sich um das Speichern und Lesen aus der Datenbank, diese Aufgabe wurde mittels den Python WebFramework CherryPy umgesetzt. Der Client Teil kümmert sich hingegen um die Darstellung sowie um die Eingaben, dies wurde mittels HTML, CSS und JavaScript umgesetzt.

### 2.1 CherryPy

Als erstes wurde eine Klasse CRUD erstellt, diese wird aufgerufen sobald man auf localhost mit der entsprechenden Port Nummer zugreift. Die Klasse öffnet die index.html Datei und sendet den Inhalt an den Browser zurück.

```
1 class CRUD(object):
    @cherrypy.expose
3     def index(self):
        return open('index.html')
```

Listing 1: Klasse zur Darstellung der Index.html

Nun wurde eine Klasse erstellt, welche auf die HTML Befehle reagiert, dies sind GET, POST, PUT, ... . In unserem Fall haben wir nur POST verwendet mit 2 Parameter. Der erste Parameter param gibt an welche Aktion wir durchführen wollen, der zweite Parameter input gibt liefert zusätzliche Informationen, wie zum Beispiel eingeben Werte.

```
@cherrypy.expose
2 class CRUDWebService(object):
4     @cherrypy.tools.accept(media='text/plain')
    def POST(self, param, input):
```

Listing 2: Klasse zur verwaltung der HTML Befehle

```
1 if param == "read":
    with sqlite3.connect(DB_STRING) as c:
3         r = c.execute("SELECT * FROM benutzer")
        response = "<table width='100%' class='table table-striped table-bordered' \
5                 ><tr><td>Nr</td><td>Vorname</td><td>Nachname</td><td>Username</td></tr>"
7         while True:
            row = r.fetchone()
9             if row is None:
                break
11            response += "<tr><td>" + str(row[0]) + "</td><td>" + row[1] + "</td><td>" + row[2] + "</td><td>" + row[3] + "</td></tr>"
13            response += "</table>"
        return response
```

Listing 3: Auslesen aller Benutzer aus der Datenbank

Hier sollen die Schritte der Laborübung erläutert werden. Alle Fragestellungen der Lehrkraft müssen hier beantwortet werden. Etwaige Probleme bzw. Schwierigkeiten sollten ebenfalls hier angeführt werden.

Es kann gut möglich sein, dass Lehrkräfte hier auch noch andere Eckpunkte explizit verlangen. Diese können dann in der selben Hierarchiestufe wie die *Umsetzung* eingeordnet werden. Viel Spass nun mit einer kleinen Übersicht von L<sup>A</sup>T<sub>E</sub>X-Elementen.

## 2.2 Tabelle

Header	Kopf
<b>Lorem</b>	Ipsum dolor sit amet, consetetur sadipscing elitr
<b>Ipsum</b>	At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus
<b>Dolor</b>	Consetetur sadipscing elitr, sed diam nonumy

Tabelle 1: Lorem ipsum dolor sit amet [1]

## 2.3 Aufzählung

- **Lorem ipsum:** dolor sit amet, consetetur sadipscing elitr
- sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat
- ut labore et dolore magna aliquyam erat, sed diam voluptua

## 2.4 Code

At vero eos et accusam et justo duo dolores et ea rebum.

Listing 4: Implizite Transaktion

## Literatur

- [1] A.S. Tanenbaum and M. Van Steen. *Verteilte Systeme: Prinzipien und Paradigmen*. Pearson Studium. Addison Wesley Verlag, 2007.

## Tabellenverzeichnis

1	Lorem ipsum dolor sit amet [1] . . . . .	3
---	--	---

## Listings

1	Klasse zur Darstellung der Index.html . . . . .	2
2	Klasse zur verwaltung der HTML Befehle . . . . .	2
3	Auslesen aller Benutzer aus der Datenbank . . . . .	2
4	Implizite Transaktion . . . . .	3

## Abbildungsverzeichnis