

# LU03c - Webservice mit Parametern



Viele Webservices benötigen beim Aufruf Parameter um die gewünschte Verarbeitung durchzuführen.

Nur wenige Webservices brauchen keine Angaben zur Verarbeitung. Viel häufiger muss der Aufrufer Parameter mitgeben, wie ...

- ... den eindeutigen Schlüssel der gewünschten Ressource
- ... welche Daten in einer Ressource zu speichern sind

Dabei unterscheiden wir zwischen verschiedenen Möglichkeiten zur Übermittlung der Parameter.

## Query-Parameter

Query-Parameter werden in der URI mitgegeben. Sie bestehen immer aus einem Schlüssel und dem entsprechenden Wert.

Die URI <https://moodle.bzz.ch/mod/lesson/view.php?id=8990&pageid=2390> enthält zwei Parameter:

- id=8990
- pageid=2390

## Umsetzung

Einerseits geben wir die Query-Parameter bei der Definition der Ressource in `app.py` an. Zum Beispiel benötigt die Ressource `EventService` eine `event_uuid`, um den gewünschten Event zu lesen:

```
api.add_resource(EventService, '/event/<event_uuid>')
```

Bei der Methode `get` geben wir diesen Query-Parameter ebenfalls an:

```
def get(self, event_uuid):
```

- Bei der Definition der Ressource wird der Parameter zwischen spitzen Klammern (`<...>`) geschrieben.
- Der Bezeichner muss bei der Definition und in der Methode identisch sein.

## Formular-Parameter

Benutzereingaben in einem Formular können theoretisch auch in der URI übertragen werden. Allerdings sieht man dann im Browserverlauf, welche Eingaben gemacht wurden. Das ist spätestens bei Login-Formularen (Passwort!!) sehr schlecht. In der Regel übertragen wir die Benutzereingaben im

Datenteil (Body) des Requests.

## Umsetzung

Für dieses Beispiel verwende ich die Bibliothek

[<https://flask-restplus.readthedocs.io/en/stable/parsing.html>][reqparse]] von Flask-RESTful.

Zunächst müssen wir die Formular-Parameter definieren. Dies geschieht in der Methode `__init__(self)` der Klasse.

```
from flask import make_response
from flask_restful import Resource, reqparse

class EventService(Resource):
    def __init__(self):
        self.parser = reqparse.RequestParser()
        self.parser.add_argument('event_name', location='form',
default=None, help='name')
        self.parser.add_argument('start', location='form', default=None,
help='start date/time')
```

In den einzelnen Methoden können wir dann die Parameter einlesen (parsen). Die Funktion `parse_args()` liefert einen Namespace (ähnlich wie ein Dictionary) mit allen Parametern. Anschliessend können wir mit den Schlüssel auf die Parameter zugreifen.

```
class EventService(Resource):
    ...
    def post(self):
        args = self.parser.parse_args()
        name = args.event_name
        start = datetime.strptime(args.start, '%Y%m%d %H%M')
```

M321-LU03



Marcel Suter

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
<https://wiki.bzz.ch/modul/m321/learningunits/lu03/parameter>

Last update: **2024/03/28 14:07**

