



## Interoperabilität

Workflow Systems and Technology  
Faculty of Computer Science  
University of Vienna  
Stefanie Rinderle-Ma, Juergen Mangler

21.05.2015

### 5. Übung: Choregraphien (12 Punkte)

Zu verwenden:

- <http://cpee.org>
- [almighty.cs.univie.ac.at](http://almighty.cs.univie.ac.at)
- <http://wwwlab.cs.univie.ac.at>

#### Aufgabe 5-1: Phase 1 – Vorbereitung (4 Punkte)

Sie besitzen ein Unternehmen, das Teile produziert. Beantworten Sie folgende Fragen:

- Was produzieren Sie (siehe Wiki)?
- Aus welchen Einzelteilen besteht der Teil (mindestens 7 Teile)? (1 JSON File das das Lager beschreibt, inklusive Anzahl der verfügbaren Teile)
- Wie sieht eine Bestellung eines Endproduktes aus (mindestens 4 Parameter)? (1 JSON File das die Bestellung beschreibt).
- 7 Fortschrittsberichte während das Endprodukt produziert wird. (1 JSON File pro Fortschritt das den Fortschritt beschreibt).

Schreiben sie 3 REST Services:

- Einzelne Teile ins Lager LEGEN (PUT), und das Lager abfragen (GET).
- Eine Bestellung abfragen (randomized parameters) (GET).
- Fortschrittsbericht abfragen (welchen Bericht als integer parameter).

Abzugeben: 1ZIP, die JSON Files, den Sourcecode der REST Services.

Registrieren der Services unter der URL im Wiki.

#### Aufgabe 5-2: Phase 2 – Correlation (6 Punkte)

To be revealed.

#### Aufgabe 5-3: Phase 3 – Annotation (2 Punkte)

To be revealed.

## Bestellung abfragen (fuer Smartphone)

Wer diese Loesung 1:1 verwendet, ist durchgefallen.

Eine Moeglichkeit

```
{
  display: 6.1,
  prozessor: "arm7",
  fingerabdrucksensor: true,
  akku: "1200mAh"
}
```

Eine zweite Moeglichkeit

```
{
  display: 4.8,
  prozessor: "arm7",
  fingerabdrucksensor: true,
  akku: "1200mAh"
}
```

Bitte Zufallswerte verwenden.

Abfragen z.b. ueber <http://wwwlab.cs.univie.ac.at/~a9506264/bestellung.php>

Letzte Änderung: 22.05.2015, 11:33 | 55 Worte

## Lagerverwaltung

### Lagerverwaltung

- Liste der Teile
- Details abfragen
- Details fuer einzelne Komponenten abfragen
- Details fuer einzelne Komponenten setzen

### Liste der Teile

Abfragen z.b. ueber <http://wwwlab.cs.univie.ac.at/~a9506264/lagerverwaltung.php>

Ergebnis:

```
{
  displays: {
    "7.1": 35,
    "5.0": 2,
    "4.8": 17
  },
  akku: {
    "1200mAh": 37,
    ...
  },
  ...
}
```

### Details abfragen

Abfragen z.b. ueber <http://wwwlab.cs.univie.ac.at/~a9506264/lagerverwaltung.php/displays>

Ergebnis:

```
{
  "7.1": 35,
  "5.0": 2,
  "4.8": 17
}
```

### Details fuer einzelne Komponenten abfragen

Abfragen z.b. ueber <http://wwwlab.cs.univie.ac.at/~a9506264/lagerverwaltung.php/displays/7.1>

Ergebnis:

```
35
```

### Details fuer einzelne Komponenten setzen

PUT auf <http://wwwlab.cs.univie.ac.at/~a9506264/lagerverwaltung.php/displays/7.1>

Ergebnis: (keines)

Client um zu putten:

```
<?php
$data = http_build_query(
    array(
        'wert' => '35'
    )
);
$options = array('http' =>
    array(
        'method' => 'PUT',
        'header' =>
            "Content-type: application/x-www-form-urlencoded\r\n" .
            "accept: application/json\r\n"
        ,
        'content' => $data
    )
);
header('content-type: text/plain');
$content = stream_context_create($options);
$result =
    file_get_contents('http://wwwlab.cs.univie.ac.at/~a9506264/lagerverwaltung.php/displays/7.1', false, $content);

print_r($result);
?>
```

Wichtiger: die Daten liegen ja in einem File (im selben Verzeichnis wie lagerverwaltung.php), das z.b. lagerverwaltung.json heisst. Damit ich in dieses File schreiben kann muessen die Rechte am almighty.cs passen. Also mit ssh einloggen und **chmod 666 lagerverwaltung.json**.