

Übung 4

Erweitern Sie die Anwendung aus Übung 3 in dem Sie die folgenden zwei, in der LV besprochenen, Features hinzufügen. Setzen Sie dabei auf die gemeinsame Planung auf. Entwickeln Sie gerne auch gemeinsam in den Teams, die Sie in der Einheit gebildet haben. Machen Sie in diesem Fall in den Kommentaren darauf aufmerksam.

Sie werden die Datenbankklasse für Update, Insert und Delete erweitern müssen.

1. Feature: Gebäude Struktur erzeugen

Die Anwendung soll ermöglichen, dass man per REST-Schnittstelle auf den diversen Ebenen Einträge erzeugt um die Struktur des Projektes mit Stockwerken, Räumen, etc. anlegen, bearbeiten und löschen kann.

Schnittstelle:

index.php

- POST Parameter JSON Objekt: data

```
{
  "action": "create", //oder "update", oder "delete"
  "listtype": "FLOORS", //oder eben die anderen
  "parentid": 1,
  "itemid": 1, //optionaler Parameter, für update und
  //delete um den Eintrag auf der gewünschten Ebene zu
  //identifizieren.
  "specification": {
    "name": "Erdgeschoss",
    "count_from_basment": 1 //oder eben die notwendigen
    //anderen felder
  }
}
```

z.b. wie oben. Die Properties auf der Hauptebene bleiben gleich.

Der Parameter "Specification" variiert je nach Ebene und erwartet alle möglichen Felder die aus der DB-Table hervorgehen (Exkl. der Timestamps)

Output:

Als Antwort wird ein JSON zurückgeliefert der angibt, ob die Aktion erfolgreich war, oder ob ein Fehler aufgetreten ist und wie die neu angelegte ID ist.

```
{
  "status": "OK", //oder "NOK", bei Fehlern
  "newId": int//integer wert der ID
}
```

2. Feature: Einkaufsliste erzeugen

Die Einkaufsliste enthält eine Liste aller Geräte die in dem Projekt verplant wurde.

Gleiche Geräte sollen zusammengezählt werden (Gleich sind sie wenn Sensoren- und Gerätenamen gleich lauten). Erzeugen Sie diesen Fall in Ihrer Datenbank.

Schnittstelle:

index.php

- POST Parameter JSON Objekt: data

```
{
    "action": "get-shoppinglist"
}
```

Output:

```
{
    "project": "<Projektname>",
    "shoppingList": [
        {
            "deviceName": "<Gerätename>",
            "count": "<Anzahl der Geräte>",
            "sensors":
            [
                { "name": "<Sensorname>" }
                // ...
            ]
        }
        // ...
    ]
}
```

Weitere Informationen

- Um die POST calls zu ermöglichen habe ich im Moodle ein Mini-Formular bereitgestellt, welches Sie in den Projektordner kopieren und anstatt der index.php in Ihrem Browser aufrufen. Sie können dort die JSON Objekte erzeugen. Verwenden Sie <https://jsonlint.com/> zum validieren Ihrer Objekte.
- Um SQL Statements zu bauen, probieren Sie diese zuerst im PHP MyAdmin aus bis sie funktionieren und übernehmen Sie diese dann erst in PHP. Damit sind Sie beim debuggen schneller.
- Versuchen Sie Code möglichst generisch zu erzeugen um Funktionalität mehrfach zu verwenden.
- Geben Sie wieder mehr acht bei der Namensgebung
- Versuchen Sie ifelse-elseif Verkettungen zu vermeiden

- Vorsicht bei MySQL DELETE: Verwenden Sie LIMIT 1 am ende des Query, um zu vermeiden, dass Sie bei Fehlern die ganze Table löschen UND machen Sie regelmäßig Backups.
- Für diese Übung ist länger Zeit, dafür ist auch mehr zu erledigen und es gibt die doppelte Anzahl Punkte zu erreichen.
- Wir werden in der nächsten Einheit über Ihren Stand der Dinge sprechen, also fangen Sie bereits vor der nächsten Einheit an zu arbeiten.

Nützliche Links

- Klassen und Objekte in PHP: <http://php.net/manual/de/language.oop5.php>
- PhP und POST Parameter: <http://php.net/manual/de/reserved.variables.post.php>
- MySQL Delete: <https://dev.mysql.com/doc/refman/8.0/en/delete.html>
- MySQL Update: <https://dev.mysql.com/doc/refman/8.0/en/update.html>
- PDO und Update: https://www.w3schools.com/php/php_mysql_update.asp
- PDO und Delete: https://www.w3schools.com/php/php_mysql_delete.asp

Wenn Sie in der Musterlösung zu Übung 3 im Listcontroller die Route Methode wie folgt verändern, können Sie das POSTCaller Skript gleich testen:

```
public function route(){

    $jsonInput = filter_input(INPUT_POST, 'data');
    $data = json_decode($jsonInput);

    $this->listType = $data->listtype;
    $parentId = $data->parentid;

    $this->createRequestedList($parentId);

    $this->formatAndDisplayData();
}
```

Input Beispiel:

```
{
    "listtype": "ROOMS",
    "parentid": 1
}
```