Ändern eines bestimmten Datensatzes - PUT

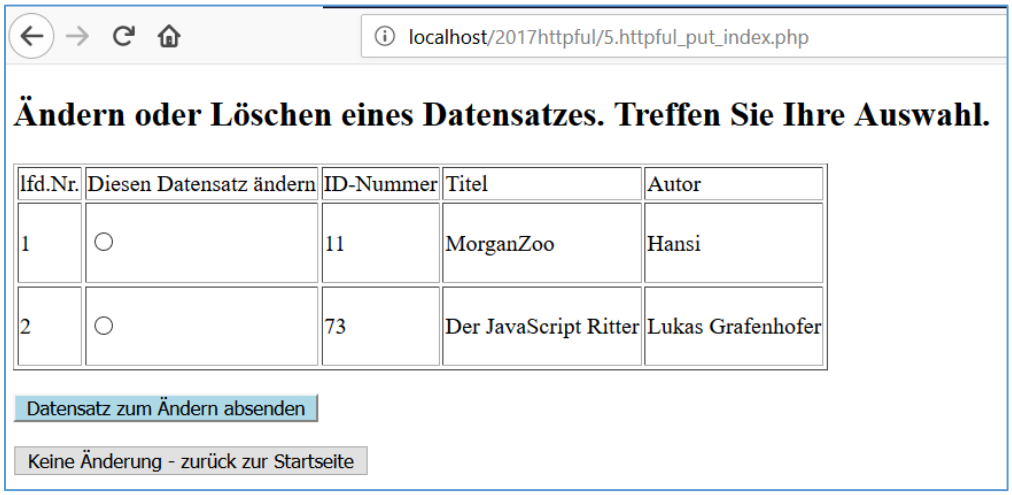
Um einen einzelnen Datensatz zu ändern, muss dieser zuvor identifiziert werden. Das ist besonders dann gut möglich, wenn auf einem Feld der Tabelle ein Primärschlüssel liegt.

Folgende Vorgangsweise ist zu empfehlen:

- 1. Dem Benutzer werden alle Datensätze angezeigt.
- Er wählt den Datensatz aus, den er ändern möchte.
- 2. Der gewählte Datensatz wird in einem Formular angezeigt.
- Der Benutzer gibt die Änderungen ein und führt sie aus.

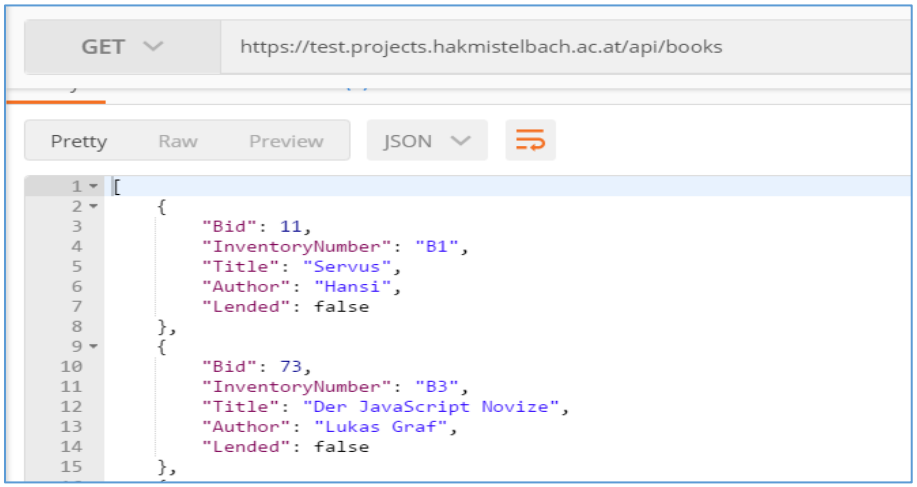
Ziel:

Alle „books“ auflisten und ändern können. Dafür soll in der Tabelle ein Radio-Button zum Anklicken angezeigt werden. Ganz vorne ist eine fortlaufende Nummerierung, die aber sonst keinen Effekt hat.



Abfrage aller vorhandenen Bücher, die am Testserver liegen unter
https://test.projects.hakmistelbach.ac.at/api/books

Ergebnis mit Postman:



1)Start: Erstelle die Datei „5.httpful_put_index.php“ im <htdocs>-Ordner von Xampp:

```
<!doctype html>
<html> <head><meta charset="utf-8">
<title>put httpful</title> </head>
<body>
<h2>Ändern oder Löschen eines Datensatzes. Treffen Sie Ihre Auswahl.</h2>
<?php
include('./httpful.phar');
$uri = "https://test.projects.hakmistelbach.ac.at/api/books";
$response = \Httpful\Request::get($uri)
->authenticateWith('demo', 'demo')
->send();
echo '<form action="5.httpful_put.php" method="post">';
echo "<table border='1'>";
echo "<tr><td>lfd.Nr.</td><td>Diesen Datensatz ändern</td><td>ID-Nummer</td><td>Titel</td>
<td>Autor</td> <td>Löschen</td></tr>"; //das ist die Headline im Kopf der Tabelle
$i = 1; //benötigt für die fortlaufende Nummerierung
foreach($response->body as $val) { //hier beginnt die Abfrage
echo '<tr><td>' . $i . '</td>'; //fortlaufende Nummerierung
$i++;
$zahl = $val -> Bid; //Bid als Variable, damit sie im value funktioniert
    echo '<td>';
    echo '<p><input type="radio" name="auswahl" value="'.$zahl.'"> </p>';
    //der name und der value des radiobuttons wird in "put" gebraucht
    echo '</td>';
echo '<td>';
echo '<p>' .  $val -> Bid  . '</p>';
echo '</td>';
echo '<td>';
    echo '<p>' .  $val -> Title  . '</p>';
echo '</td>';
echo '<td>';
    echo '<p>' .  $val -> Author  . '</p>';
echo '</td></tr>';
}
echo "</table>";
echo '<p><input type="submit" value="Datensatz zum Ändern absenden" style="background-
color:lightblue;"></p>';
echo '<p><a href="#"><button type="button">Keine Änderung - zurück zur Startseite</button></a>
</p>';
echo "</form>";
?>
</body></html>
```

Code:

```
<!doctype html>
<html> <head><meta charset="utf-8">
<title>put httpful</title> </head>
<body>
<h2>Ändern oder Löschen eines Datensatzes. Treffen Sie Ihre Auswahl.</h2>
<?php
include('./httpful.phar');
$uri = "https://test.projects.hakmistelbach.ac.at/api/books";
$response = \Httpful\Request::get($uri)
->authenticateWith('demo', 'demo')
->send();
echo '<form action="5.httpful_put.php" method="post">';
echo "<table border='1'>";
echo "<tr><td>lfd.Nr.</td><td>Diesen Datensatz ändern</td><td>ID-Nummer</td><td>Titel</td>
<td>Autor</td></tr>"; //das ist die Headline im Kopf der Tabelle
$i = 1; //benötigt für die fortlaufende Nummerierung
foreach($response->body as $val) { //hier beginnt die Abfrage
```

```
echo '<tr><td>' . $i . '</td>'; //fortlaufende Nummerierung
$i++;
$zahl = $val -> Bid; //Bid als Variable, damit sie im value funktioniert
  echo '<td>';
   echo '<p><input type="radio" name="auswahl" value="'.$zahl.'"> </p>';
   //der name und der value des radiobuttons wird in "put" gebraucht
   echo '</td>';
echo '<td>';
echo '<p>' .  $val -> Bid  . '</p>';
echo '</td>';
echo '<td>';
          echo '<p>' .  $val -> Title  . '</p>';
echo '</td>';
echo '<td>';
          echo '<p>' .  $val -> Author  . '</p>';
echo '</td></tr>';
}
echo "</table>";
echo '<p><input type="submit" value="Datensatz zum Ändern absenden" style="background-
color:lightblue;"></p>';
echo '<p><a href="#"><button type="button">Keine Änderung - zurück zur
Startseite</button></a></p>';
echo "</form>";
?>
</body></html>
```

Ergebnis mit einem Browser:

Info:

In Zeile 36 wurde zur optischen Hervorhebung ein kleiner CSS-Inline Befehl verwendet.

Beachte 1:

In Zeile 20 wird die „ID“, die hier „Bid“ heißt, einer Variablen („$zahl“) übergeben, damit sie nicht als String, sondern als echte Zahl in den „value“ des Radiobuttons geschrieben werden kann. Zusätzlich wird dem Radio-Button ein Name gegeben.

```
20  $zahl = $val -> Bid; //Bid als Variable, damit sie im value funktioniert
21      echo '<td>';
22      echo '<p><input type="radio" name="auswahl" value="'.$zahl.'"> </p>';
23      //der name und der value des radiobuttons wird in "put" gebraucht
24      echo '</td>';
```

Dieser wird mit $_POST[„auswahl“] in der darauffolgenden Datei „httpful_put.php“ übernommen und ebenfalls in eine Variable geschrieben. Diese kann dann als ausgewählte „ID“ an die URI gehängt werden.

```
 7  <?php
 8  $test = $_POST['auswahl'];
 9  include('./httpful.phar');
10  $uri = "https://test.projects.hakmistelbach.ac.at/api/books/$test";
11  $response = \Httpful\Request::get($uri)
```

Beachte 2:

Alle Anweisungen sind bequem in „echo“ untergebracht. Vom <table border> bis zum <form> und <submit-Button>. Wichtig sind nur die Anführungszeichen und der Strichpunkt am Ende.

2) zum Ändern eines Datensatzes

Der ausgewählte Datensatz wird mit allen Daten innerhalb eines Formulars angezeigt. Das Ergebnis umfasst genau diesen einen Datensatz. Die aktuellen Inhalte der Felder aus diesem Datensatz werden innerhalb der Eingabefelder des Formulars angezeigt.

Der Benutzer kann die Inhalte nach Belieben ändern. Aus bestimmten Gründen wird die ID (hier „Bid“) nicht zum Ändern angeboten, da sie ja beim Anlegen automatisch erstellt wurde. Beim Absenden werden die Inhalte der Eingabefelder an die Datei „5.httpful_put2.php“ übermittelt.

Info: Zum Beweis und zur Info lasse unten die ID, hier bei uns genannt Bid, mitlaufen und ausgeben. Der Hauptgrund dafür ist, dass ein „name“ und ein „value“ erzeugt werden. Diese werden nämlich unbedingt benötigt, um diese ID, die ja im Radio-Button angeklickt wurde, zu bewahren und auf die nächste PHP-Datei übertragen zu können. Man benötigt unbedingt ein Formular <form>. Nur diese beinhaltet eine „action“ – welche „name“ und „value“ von <input>-Feldern übernehmen kann durch „$_POST[]“. Das kann ein <input>-Feld für Radio-Buttons (type=“radio“) sein, oder sogar vom Typ „type=“hidden“, oder wie bei uns ohne „type“. Hauptsache ein „input“-Feld im Formularkörper.

Erstelle eine Datei „5.httpful_put.php“:


```php
<!doctype html>
<html>
<head><meta charset="utf-8">
<title>update httpful</title>
</head>
<body>
<?php
$test = $_POST['auswahl'];
include('./httpful.phar');
$uri = "https://test.projects.hakmistelbach.ac.at/api/books/$test";
$response = \Httpful\Request::get($uri)
->sendsJson()
->authenticateWith('demo', 'demo')
->send();
echo "<h3>Bitte Änderungen vornehmen.</h3>";
echo '<form action="5.httpful_put2.php" method="post">';
//Aufzählung aber ohne Bid - damit diese nicht geändert wird, ist ja automatisch
echo '<p><input name="invnr" value="'.$response->body->InventoryNumber.'">Inventurnummer</p>';
echo '<p><input name="titel" value="'.$response->body->Title.'">Titel</p>';
echo '<p><input name="autor" value="'.$response->body->Author.'">Autor</p>';

echo '<p><input type="submit" value="Änderung absenden"></p>';
echo '<p><input type="reset"></p>';
echo '<p>Nur zur Info:</p>';
echo '<p><input name="auswahl_uebergabe" value="'.$test.'">benutzte ID-Nummer</p>';
//in Zeile 25 wird ein name und value noch in der form erstellt, zur Übergabe bei action
echo "</form>";
?>
</body></html>
```

Code:

```
<!doctype html>
<html>
<head><meta charset="utf-8">
<title>update httpful</title>
</head>
<body>
<?php
$test = $_POST['auswahl'];
```

```
include('./httpful.phar');
$uri = "https://test.projects.hakmistelbach.ac.at/api/books/$test";
$response = \Httpful\Request::get($uri)
->sendsJson()
->authenticateWith('demo', 'demo')
->send();
echo "<h3>Bitte Änderungen vornehmen.</h3>";
echo '<form action="5.httpful_put2.php" method="post">';
//Aufzählung aber ohne Bid - damit diese nicht geändert wird, ist ja automatisch
echo '<p><input name="invnr" value="'.$response->body-
>InventoryNumber.'">Inventurnummer</p>';
echo '<p><input name="titel" value="'.$response->body->Title.'">Titel</p>';
echo '<p><input name="autor" value="'.$response->body->Author.'">Autor</p>';

echo '<p><input type="submit" value="Änderung absenden"></p>';
echo '<p><input type="reset"></p>';
echo '<p>Nur zur Info:</p>';
echo '<p><input name="auswahl_uebergabe" value="'.$test.'">benutzte ID-Nummer</p>';
//in Zeile 25 wird ein name und value noch in der form erstellt, zur Übergabe bei action
echo "</form>";
?>
</body></html>
```

3) Daten werden in der Datenbank geändert

Erstelle eine Datei „5.httpful_put2.php“.

Durch das Absenden der vorigen Datei, 5.httpful_put.php“ wird diese neue PHP-Datei aufgerufen, 5.httpful_put2.php“.

Dabei wird die „put“-Methode angewendet, die für eine bestimmte ID einzelne Daten ändern lässt. Die ID (hier BID) wird nicht in der „uri“ aufgerufen, sondern im „body“:

Nicht hier in Zeile 9:

```
 7  <?php
 8  include('./httpful.phar');
 9  $uri = "https://test.projects.hakmistelbach.ac.at/api/books/";
10  $response = \Httpful\Request::put($uri)
```

Sondern hier im <body>:

```
14  ->body ('{
15      "Bid": "'.$_POST['auswahl_uebergabe'].'",
16      "InventoryNumber": "'.$_POST['invnr'].'",
```

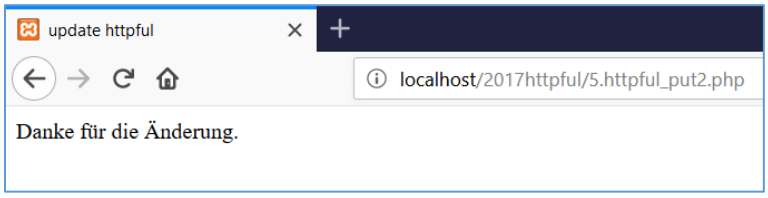
Diese $_POST holt sich die Info für die betreffende ID aus dem PHP-Dokument, das die Ausgangsdatei war. Hier die „5.httpful_put.php“. Dort wurde sie bewusst in Zeile 25 angelegt, um sie hier übergeben zu können.

```php
<!doctype html>
<html>
<head><meta charset="utf-8">
<title>update httpful</title>
</head>
<body>
<?php
include('./httpful.phar');
$uri = "https://test.projects.hakmistelbach.ac.at/api/books/";
$response = \Httpful\Request::put($uri)
->sendsJson()
->authenticateWith('demo', 'demo')
//Die ID (Bid) muss hier versorgt werden (Zeile 15) und nicht bei uri Zeile 9
->body ('{
    "Bid": "'.$_POST['auswahl_uebergabe'].'",
    "InventoryNumber": "'.$_POST['invnr'].'",
    "Title": "'.$_POST['titel'].'",
    "Author": "'.$_POST['autor'].'",
    "Lended": false
}')
->send();
echo "Danke für die Änderung.";
?>
</body></html>
```

Code:

```
<!doctype html>
<html>
<head><meta charset="utf-8">
<title>update httpful</title>
</head>
<body>
<?php
include('./httpful.phar');
$uri = "https://test.projects.hakmistelbach.ac.at/api/books/";
$response = \Httpful\Request::put($uri)
->sendsJson()
->authenticateWith('demo', 'demo')
//Die ID (Bid) muss hier versorgt werden (Zeile 15) und nicht bei uri Zeile 9
->body ('{
   "Bid": "'.$_POST['auswahl_uebergabe'].'",
   "InventoryNumber": "'.$_POST['invnr'].'",
   "Title": "'.$_POST['titel'].'",
         "Author": "'.$_POST['autor'].'",
   "Lended": false
}')
->send();
echo "Danke für die Änderung.";
?>
</body></html>
```

Ergebnis im Browser:



INFO:

`PUT` is used to **create or update**.

HTTP Verb	CRUD
POST	Create
GET	Read
PUT	Update/Replace
PATCH	Update/Modify
DELETE	Delete

http://www.restapitutorial.com/lessons/httpmethods.html

https://restfulapi.net/rest-put-vs-post/ :

Generally, in practice, always use `PUT` for UPDATE operations.	Always use `POST` for CREATE operations.

`PUT /questions/{question-id}`	`POST /questions`
`PUT` method is idempotent. So if you send retry a request multiple times, that should be equivalent to single request modification.	`POST` is NOT idempotent. So if you retry the request N times, you will end up having N resources with N different URIs created on server.

```
GET     /device-management/devices : Get all devices
POST    /device-management/devices : Create a new device

GET     /device-management/devices/{id} : Get the device information
identified by "id"
PUT     /device-management/devices/{id} : Update the device
information identified by "id"
DELETE  /device-management/devices/{id} : Delete device by "id"
```