

Abfrage und Anzeige aus der Datenbank mit httpful und REST (inkl. API)

Inhalt:

- **1) Login mit httpful und API (SLIM)**
- **2) Registrieren neuer Nutzer**
- **3) Abfrage von Terminen**
- **4) Termine inkl. Symbol zum Ändern (put)**
- **5) Termine ändern (put)**
- **6) Termine Löschen (delete)**

Name	Änderungsdatum	Typ	Größe
api	04.01.2020 23:55	Dateiordner	
css	05.01.2020 01:09	Dateiordner	
img	04.01.2020 22:41	Dateiordner	
js	05.01.2020 01:08	Dateiordner	
httpful.phar	14.01.2017 09:35	PHAR-Datei	63 KB
login1.start.html	09.01.2020 23:14	Firefox HTML Docum...	2 KB
login2.php	09.01.2020 23:02	PHP-Datei	1 KB
login3.privat.php	09.01.2020 22:12	PHP-Datei	2 KB
login4.logout.php	09.01.2020 22:46	PHP-Datei	1 KB
termine.php	09.01.2020 21:27	PHP-Datei	2 KB
user1.anlegen.html	09.01.2020 23:14	Firefox HTML Docum...	2 KB
user2.anlegen.php	09.01.2020 23:15	PHP-Datei	1 KB

1) Login mit httpful und API

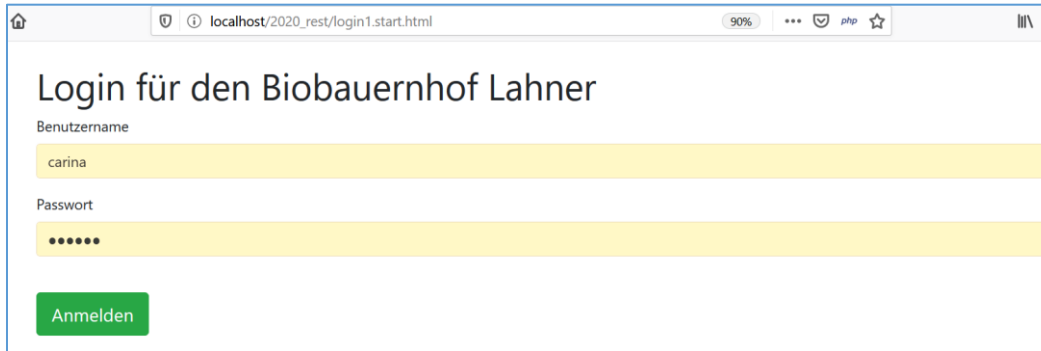
LOGIN:

1a) Datenbank: „schueler“, Tabelle: login

#	Name	Typ	Kollation	Attribute
1	I_id	int(11)		
2	benutzer	varchar(30)	utf8_general_ci	
3	passwort	varchar(50)	utf8_general_ci	

I_id	benutzer	passwort
1	carina	geheim
2	manuel	geheim

1b) Formular in html:



localhost/2020_rest/login1.start.html

Login für den Biobauernhof Lahner

Benutzername

Passwort

Anmelden

```
13 <body>
14 <section class="container">
15     <br>
16     <h1>Login für den Biobauernhof Lahner</h1>
17
18     <form name="loginform" action="login2.php" method="post">
19         <div class="form-group">
20             <label for="benutzer">Benutzername</label>
21             <input type="text" class="form-control" id="benutzer" name="bn"
22                 placeholder="Benutzername">
23         </div>
24         <div class="form-group">
25             <label for="passwort">Passwort</label>
26             <input type="password" class="form-control" id="passwort"
27                 name="pw" placeholder="Passwort">
28         </div>
29         <br>
30         <button type="submit" name="submitLogin" class="btn btn-success
31             btn-lg" style="margin-bottom:8px" >Anmelden</button>
32     </form>
33 </section>
```

Wichtig für die Weiterleitung und Weiterverarbeitung sind:

- <form>-Tag schließt das Formular ein: von Zeile 18 und 30
- „action“ und „method“ in Zeile 18
- Jeweils einen „name“ in jeder input-Zeile, die später übergeben wird an die PHP-Datei, die in „action“ angegeben ist.

1c) Die dazu passende PHP-Datei:

Auch hier erfolgen in PHP-Tags die Verbindung zum passenden Pfad der API und danach die GET-Abfrage.

```

1  <?php
2  session_start();
3  include('./httpful.phar');
4
5  $uri = "http://localhost/2020_rest/api/public/login";
6
7  $response = \Httpful\Request::get($uri)
8  ->expectsJson()
9  ->send();

```

Darunter die Abfrage, ob die LOGIN-Daten übereinstimmen:

Erklärungen:

- In den Zeilen 11 bis 14 wird geschaut, ob überhaupt Daten in die <input>-Felder des Formulars eingegeben wurden. Es würde eine Fehlermeldung mit „echo“ erscheinen. Dabei in Zeile 12: mit \$_POST wird genau der Name geholt, der auch im HTML-Formular beim SUBMIT-Button als Name hinterlegt ist.
- Das „foreach“ folgt und überprüft in einer „if“-Verzweigung, ob die im Formular eingegebenen Daten (bn und pw) mit den Daten aus der Datenbank übereinstimmen z.B.: \$_POST['bn']==\$val -> benutzer
- Zeile 21-22: Wenn die Benutzereingabe erfolgreich war erfolgt die Weiterleitung auf das Dokument, das ja geschützt sein soll. Diese Weiterleitung passiert, indem man einen „Location-header“ aufruft. Ein „header“ ist ähnlich einem Cookie eine Steuerinformation für den Browser.

```
header('Location: login3.privat.php');
```

Nach dieser „Umleitung“ muss aber das Skript beendet werden, nämlich mit „exit“:

```
exit;
```

- In Zeile 24 kann man nicht mit einem einfachen „echo“ die Falscheingabe eines Nutzers ausgeben, da mit der „foreach“-Schleife genauso viele „Falscheingabe-Meldungen“ ausgegeben werden würden, wie Elemente vorhanden sind. Daher erfolgt die Ausgabe über die Variable \$istEingabefehlerhaft, die bis zur Zeile 26 wirkt.

```

else {$istEingabefehlerhaft="true"; }
}if($istEingabefehlerhaft="true")
echo "Die eingegebenen Daten sind falsch";

```

```

11 $istEingabefehlerhaft="";
12 if(isset($_POST["submitLogin"])) //Überprüfung, dass Submitbutton
   gedrückt
13 {if(empty($_POST["bn"]) OR empty($_POST["pw"]))
14 {echo "Beide Felder werden benötigt.";}
15 else {
16

```

```

16
17 ▼ foreach ($response->body as $val) {
18     if ($_POST['bn']==$val -> benutzer && $_POST['pw']==$val ->
        passwort)
19 ▼         {
20             $_SESSION['benutzername']= $_POST['bn'];
21             header('Location: login3.privat.php');
22             exit;
23
24         } else {$istEingabefehlerhaft="true";
25         }
26 }if($istEingabefehlerhaft="true")
27 echo "Die eingegebenen Daten sind falsch";
28 }}
29 ?>

```

Code:

```

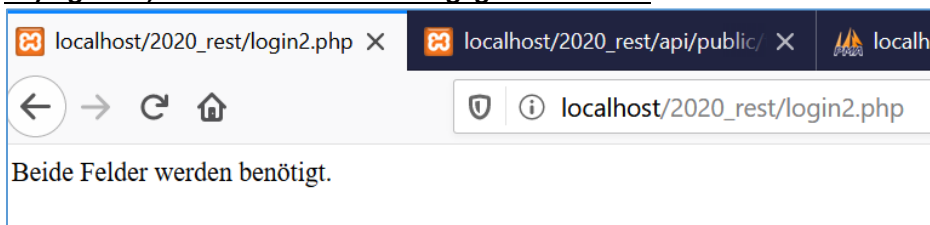
$istEingabefehlerhaft="";
if(isset($_POST["submitLogin"])) //Überprüfung, dass Submitbutton gedrückt
{if(empty($_POST["bn"]) OR empty($_POST["pw"]))
{echo "Beide Felder werden benötigt.";}
else {

foreach ($response->body as $val) {
    if ($_POST['bn']==$val -> benutzer && $_POST['pw']==$val -> passwort)
    {
        $_SESSION['benutzername']= $_POST['bn'];
        header('Location: login3.privat.php');
        exit;

    } else {$istEingabefehlerhaft="true";
    }
}if($istEingabefehlerhaft="true")
echo "Die eingegebenen Daten sind falsch";
}}
?>

```

1d)Ergebnis, wenn falsche Daten eingegeben werden



1e) Geschützte Seite



Werden die beiden Daten „benutzer“ und „passwort“ korrekt eingegeben kommt man auf die geschützte Seite weiter.

Dies ist ebenfalls eine PHP-Seite, aber ohne httpful und API. Nur reiner Bootstrap-Code.

```
10 <link href="css/bootstrap.css" rel="stylesheet">
11 </head>
12
13 <body>
14 <section class="container">
15     <br><br>
16     <h1>Du hast das Login geschafft und bist im geschützten Bereich. </h1>
17 <?php
18     session_start();
19
20     $istBenutzerAngemeldet = isset($_SESSION['benutzername']);
21     if ($istBenutzerAngemeldet) {
22         $benutzername = $_SESSION['benutzername'];
23     } else {
24         $benutzername = '';
25     }
26     if ($istBenutzerAngemeldet):
27     ?>
```

```
28
29     <p>Du bist angemeldet als: <?= htmlspecialchars($benutzername); ?>
30     </p>
31     <p>Weitere vertrauliche Inhalte ...</p>
32     <p><a href="login4.logout.php">Abmelden</a></p>
33     <?php else: ?>
34     <p>Zugriff nur für angemeldete Benutzer.</p>
35     <p><a href="login1.start.html">Zur Anmeldung</a></p>
36     <?php endif; ?>
37 </section>
38 </body>
39 </html>
```

Zeile 20: Nun muss in der „login3.privat.php“ geprüft werden, ob der Benutzer angemeldet ist. Dazu muss man das Element „benutzername“ in der Session-Variablen betrachten: ist der „benutzername“ vorhanden, kann man davon ausgehen, dass der Login geklappt hat.

Der gesamte Vorgang, inkl. „isset“ wird in einer Variablen „istBenutzerAngemeldet“ gespeichert

```
$istBenutzerAngemeldet = isset($_SESSION['benutzername']);
```

Vorher muss die Session gestartet sein, hier in Zeile 18:

```
session_start();
```

Die Anwendung der „erfolgreichen Anmeldung“ wird in der Zeile 26 verwendet:

```
<?php if ($istBenutzerAngemeldet): ?>
```

Damit, wie in Zeile 29 erkennbar, auch direkt der Name des Benutzers ausgegeben wird, wird zuerst in Zeile 21 geprüft, ob der Benutzer angemeldet ist:

```
if ($istBenutzerAngemeldet) {  
    $benutzername = $_SESSION['benutzername'];  
} else {  
    $benutzername = ""; }  

```

Darin erhält die Variabel „\$benutzername“ die Information der Session, in der der Benutzername gespeichert ist. Ist kein Benutzer angemeldet, soll die Variable mit einem leeren String belegt werden.

Ausgegeben wird dies in der Zeile 29 in einem kurzen php-Tag:

```
<p>angemeldet als <?= htmlspecialchars($benutzername); ?></p>
```

1f)logout

```
1 <?php  
2 session_start();  
3 unset($_SESSION['benutzername']);  
4 header('Location: login1.start.html');  
5 exit;  
6 ?>
```

In Zeile 3 geht man davon aus, dass in der Session-Variablen der Benutzername gesetzt war.

```
unset($_SESSION['benutzername']);
```

„unset“ löscht das Element „benutzername“. Wenn es nicht gesetzt war, weil ein Besucher „logout“ aufgerufen hat, ohne vorher angemeldet gewesen zu sein, ist das kein Problem.

```
header('Location: login1.start.html');  
exit;
```

„header“ vollzieht eine Umleitung auf den „login“

2)Registrieren eines neuen Users

Die „login1.start.html“ wird um einen Button „Registrieren“ erweitert: Zeile 33-35

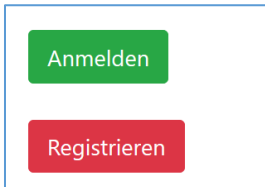
```
32 <br>
33 <section class="container">
34 <a href="user1.anlegen.html"><button type="text" class="btn btn-
   danger btn-lg" style="margin-bottom:8px" >Registrieren</button></a>
35 </section>
36 <br>
37 <script src="js/jquery-3.3.1.min.js"></script>
```

Code:

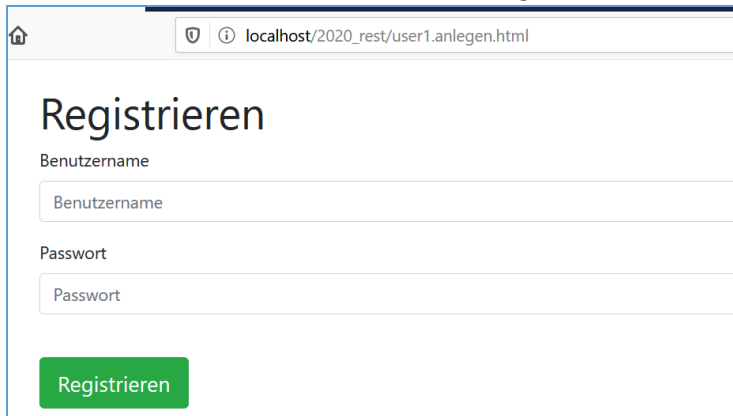
```
<section class="container">
  <a href="user1.anlegen.html"><button type="text" class="btn btn-danger btn-lg" style="margin-
  bottom:8px" >Registrieren</button></a>
</section>
```

Ergebnis:

Neuer Button Registrieren



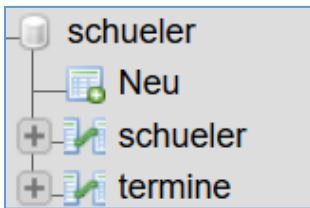
Dann kommt man auf die Seite „user1.anlegen.html“, die ein Formular darstellt.

A screenshot of a web browser window. The address bar shows 'localhost/2020_rest/user1.anlegen.html'. The page title is 'Registrieren'. Below the title, there are two input fields: 'Benutzername' and 'Passwort'. At the bottom of the form is a green button labeled 'Registrieren'.

Diese Seite leitet auf die PHP-Site, welche die Daten in die Datenbank schreibt:

```
1 <!doctype html>
2 <html>
3 <head><meta charset="utf-8"><title>user.anlegen</title></head>
4 <body>
5
6 <?php
7     include('./httpful.phar');
8     $uri = "http://localhost/2020_rest/api/public/user/add";
9     $response = \Httpful\Request::post($uri)
10    ->sendJson()
11    ->body('{
12        "benutzer": "'.$_POST['bn'].'",
13        "passwort": "'.$_POST['pw'].'"
14    }')
15    ->send();
16 ?>
17
18 <script> alert("Danke für die Registrierung. Melden Sie sich bitte im
19 LOGIN an.");
20     window.location.href = "login1.start.html";
21 </script>
22 </body>
23 </html>
```

3)Tabelle „termine“ anzeigen aus der Datenbank schueler



#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra
1	t_id	int(11)			Nein	kein(e)		AUTO_INCREMENT
2	datum	date			Nein	kein(e)		
3	inhalt	varchar(100)	utf8_general_ci		Nein	kein(e)		

	t_id	datum	inhalt
schuen	1	2020-01-31	Nachwandern ab 18 Uhr, Treffpunkt: vor dem Stall,...
schuen	2	2020-02-20	Kochseminar mit Frau Hilde - deftiges vom Bioschwe...

die Website sieht am Beginn so aus:



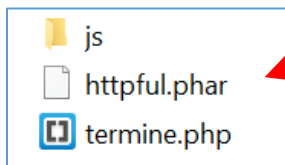
Das ist eine Site mit Bootstrap. **Liegt in XAMPP**, da es PHP verwendet.

ser PC > OS (C:) > xampp > htdocs > 2020_rest

Name	Änderungsdatum	Typ
css	04.01.2020 19:13	Dateiordner
img	04.01.2020 22:41	Dateiordner
js	04.01.2020 19:13	Dateiordner
termine.php	04.01.2020 22:43	PHP-Datei

Einbau von httpful-Elementen in die Datei „termine.php“.

Zuerst muss man die zuständige Datei „httpful.phar“ hereinkopieren.



Einbinden der HTTPful-Datei: innerhalb von PHP-Tags

```
17 <body>
18     <?php
19         include('./httpful.phar');
20     ?>
21 <section class="container">
22     <br>
23     <h1>Willkommen auf der Seite
```

Darunter folgt, ebenfalls in PHP-Tags, die Verbindung zum API-Pfad. Dann mit der GET-Abfrage, die das Format JSON haben muss.

```
30     <?php
31     $uri = "http://hier folgt die API-Adresse";
32     $response = \Httpful\Request::get($uri)
33     ->expectsJson()
34     ->send();
```

Code:

```
$uri = "http://hier folgt die API-Adresse";
$response = \Httpful\Request::get($uri)
->expectsJson()
->send();
```

Pfad der „uri“:

Die hier verwendete API liegt gleich neben den Ordnern für Bootstrap und ist somit nicht weit entfernt. Dort gibt es im Ordner „public“ die „index.php“. Darin wird der Pfad für die Abfrage aus der

Datenbank festgelegt.

User PC > OS (C:) > xampp > htdocs > 2020_rest > api		
Name	Änderungsdatum	Typ
config	04.01.2020 23:55	Dateiordner
public	05.01.2020 20:27	Dateiordner
src	31.12.2019 01:03	Dateiordner
vendor	04.01.2020 23:55	Dateiordner
composer.json	29.12.2019 16:11	JSON-Datei
composer.lock	27.12.2019 18:02	LOCK-Datei

Daher gibt es hier den Pfad:

```
"http://localhost/2020_rest/api/public/termine";
```

Wichtig: kein Slash am Ende von termine

Wichtig: ->expectsJson()

Dieser wird nun in die Datei eingesetzt:

```
30 <?php
31 $uri = "http://localhost/2020_rest/api/public/termine";
32 $response = \Httpful\Request::get($uri)
33 ->expectsJson()
34 ->send();
```

Darunter folgt an einer passenden Stelle die Tabelle, die, wiederum mit PHP-Tags, mit <echo> die Daten ausgibt. Hier wird auch eine FOREACH-Schleife verwendet.

Die Variable \$i wird in Zeile 41 auf 1 gesetzt und dann in Zeile 44 hochgezählt. Dadurch kann man in Zeile 57 ausgeben und somit die Anzahl der vorhandenen Elemente ausgeben.

```
Insgesamt Anzeigen online: 2
```

```
37 //Tabellenbeginn
38 echo "<table border='1' width='100%' class='table-striped'>";
39 echo "<tr><th>ID</th><th>Datum</th><th>Inhalt</th></tr>";
40
41 $i = 1; //benötigt für die fortlaufende Nummerierung
42 foreach($response -> body as $val)
43 {echo '<tr>';
44     $i++;
45     echo '<td>';
46         echo '<p>'. $val -> t_id . '</p>';
47     echo '</td>';
48     echo '<td>';
49         echo '<p>'. $val -> datum . '</p>';
50     echo '</td>';
51     echo '<td>';
52         echo '<p>'. $val -> inhalt . '</p>';
53     echo '</td>';
54     echo '</tr>';
55 }
56 echo "</table>";
```

```

55     }
56     echo "</table>";|
57     echo '<p>Insgesamt Anzeigen online: <span class="badge badge-primary
badge-pill">' . ($i-1) . '</span></p>';
58     ?>
59     <p>Hier folgen die Termine, die aus der Datenbank ausgelesen werden.
</p>
60     </div></div>
61 </section>

```

Code:

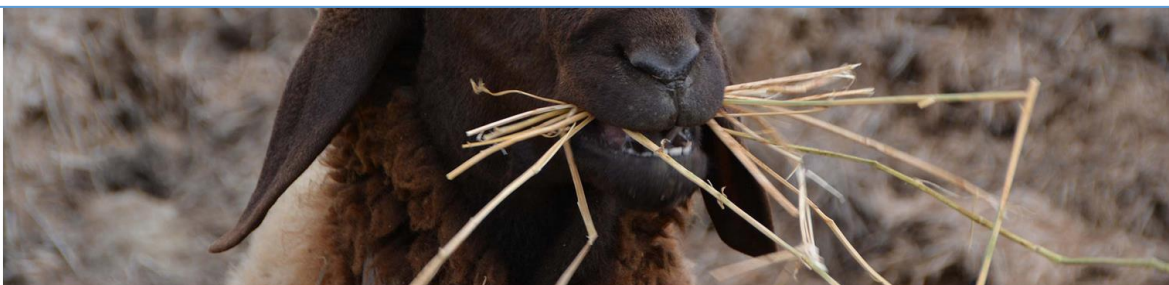
```

echo "<table border='1' width='100%' class='table-striped'>";
    echo "<tr><th>ID</th><th>Datum</th><th>Inhalt</th></tr>";

    $i = 1; //benötigt für die fortlaufende Nummerierung
    foreach($response -> body as $val)
    {echo'<tr>';
        $i++;
        echo'<td>';
            echo'<p>'. $val -> t_id . '</p>';
        echo '</td>';
        echo'<td>';
            echo'<p>'. $val -> datum . '</p>';
        echo '</td>';
        echo'<td>';
            echo'<p>'. $val -> inhalt . '</p>';
        echo '</td>';
        echo '</tr>';
    }
    echo "</table>";
    echo '<p>Insgesamt Anzeigen online: <span class="badge badge-primary badge-pill">' . ($i-1)
.</span></p>';

```

Ergebnis:



unsere Termine in den nächsten Wochen:

ID	Datum	Inhalt
1	2020-01-31	Nachtwandern ab 18 Uhr, Treffpunkt: vor dem Stall,
2	2020-02-20	Kochseminar mit Frau Hilde - deftiges vom Bioschwein

Insgesamt Anzeigen online: 2

Hier folgen die Termine, die aus der Datenbank ausgelesen werden.

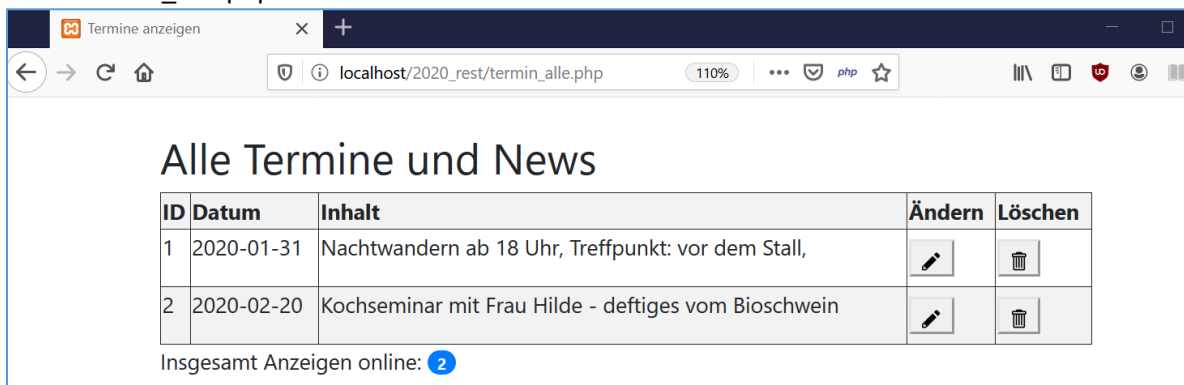
4)Übersicht anzeigen mit Symbole zum Ändern und Löschen

put – verwendet man, um Daten zu ändern
delet – zum Löschen





INFO:

- Verwende die „GET-Methode“ statt POST beim Formular
- Diese „id“ muss vom Dokument „termin_alle.php“ (mit dem Formular im Button Zeile 43-45) an das Dokument „termin_put.php“ übergeben werden, wo es auch in die API geschickt wird – dort wird es als {id} entgegengenommen.
- API SLIM: erstelle neuen Eintrag „termin/{id}“. Diese werden durch einen einzigen (single item) zugeordnet. Der „placeholder“ ist in geschwungenen Klammern {id}.

Datei: termin_alle.php

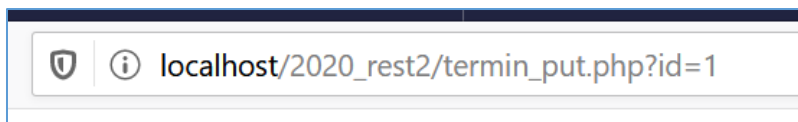


The screenshot shows a web browser window with the title "Termine anzeigen". The address bar shows "localhost/2020_rest/termin_alle.php". The main content area displays the heading "Alle Termine und News" above a table. The table has five columns: ID, Datum, Inhalt, Ändern, and Löschen. There are two rows of data. Below the table, it says "Insgesamt Anzeigen online: 2".

ID	Datum	Inhalt	Ändern	Löschen
1	2020-01-31	Nachtwandern ab 18 Uhr, Treffpunkt: vor dem Stall,		
2	2020-02-20	Kochseminar mit Frau Hilde - deftiges vom Bioschwein		

Info:

- Weiterleitung bei Klick auf „Ändern“ (Bleistiftsymbol) erfolgt in Zeile 43-45 mit einem Button der in „a href“ eingebettet ist und ganz selbständig das Formular beinhaltet: submit, formaction und name mit value.
Der „name“ wird mit der methode=“get“ (siehe Zeile 23) in der URL



The screenshot shows a browser address bar with the URL "localhost/2020_rest2/termin_put.php?id=1".

```

12 ▼ <body>
13 ▼   <div class="container">
14   <?php
15     require ("./httpful.phar");
16
17     $uri = "http://localhost/2020_rest/api/public/termine";
18     $response = \Httpful\Request::get($uri)
19     ->expectsJson()
20     ->send();
21
22     //Tabellenbeginn
23     echo '<form method="get">';
24     echo "<br><h2>Alle Termine und News</h2>";
25     echo "<table border='1' width='100%' class='table-striped'>";
26     echo "<tr><th>ID</th><th>Datum</th><th>Inhalt</th><th>Ändern</th>
27     <th>Löschen</th></tr>";
28     $i = 1;|
29     foreach($response -> body as $val)
30     {echo '<tr>';
31       $i++;
32       $id = $val -> t_id;
33       echo '<td>';
34         echo '<p>'. $val -> t_id . '</p>';
35       echo '</td>';

```

```

36         echo '<p>'. $val -> datum . '</p>';
37       echo '</td>';
38       echo '<td>';
39         echo '<p>'. $val -> inhalt . '</p>';
40       echo '</td>';
41       echo '<td>';
42       //folgt Weiterleitung mit Link - Bleistift oder Papierkorb
43       echo '<a href="termin_put.php">
44       <button type="submit" formaction="termin_put.php" name="id"
45       value="'. $id .'" ></a>
46       ';
47       echo '</td>';
48       echo '<td>';
49       echo '<a href="termin_delete.php">
50       <button type="submit" formaction="termin_delete.php"
51       name="auswahl_delete" value="'. $id .'"></a>
52       ';
53       echo '</td>';
54       echo '</tr>';
55     }
56     echo "</table>";
57     echo '<p>Insgesamt Anzeigen online: <span class="badge badge-primary badge-
58     pill">' . ($i-1) . '</span></p>';
59     echo "</form>";
60   ?> <!--Ende PHP-->
61 </div>

```

Code:

```

<div class="container">
<?php
require ("./httpful.phar");

    $uri = "http://localhost/2020_rest/api/public/termine";
    $response = \Httpful\Request::get($uri)
    ->expectsJson()
    ->send();

    //Tabellenbeginn

```

```

        echo '<form method="get">';
    echo "<br><h2>Alle Termine und News</h2>";
        echo "<table border='1' width='100%' class='table-striped'>";
        echo
"<tr><th>ID</th><th>Datum</th><th>Inhalt</th><th>Ändern</th><th>Löschen</th></tr>";
        $i = 1;
        foreach($response -> body as $val)
        {echo'<tr>';
            $i++;
            $id = $val -> t_id;
            echo'<td>';
                echo'<p>'. $val -> t_id . '</p>';
            echo '</td>';
            echo'<td>';
                echo'<p>'. $val -> datum . '</p>';
            echo '</td>';
            echo'<td>';
                echo'<p>'. $val -> inhalt . '</p>';
            echo '</td>';
        echo'<td>';
        //folgt Weiterleitung mit Link - Bleistift oder Papierkorb
        echo'<a href="termin_put.php">
        <button type="submit" formaction="termin_put.php" name="id" value="'. $id .'" ></a>
        ';
            echo'</td>';
            echo '<td>';
            echo'<a href="termin_delete.php">
            <button type="submit" formaction="termin_delete.php" name="auswahl_delete"
value="'. $id .'"></a>
            ';
                echo '</td>';
            echo '</tr>';
        }
        echo "</table>";
        echo ' <p>Insgesamt Anzeigen online: <span class="badge badge-primary badge-pill">' . ($i-1)
.'</span></p>';
        echo "</form>";
    ?> <!--Ende PHP-->
</div>

```

5) Daten Ändern

5a) termin_put.php

Nachdem man in der Tabelle „termine_alle.php“ das Symbol „Ändern“ (Bleistift) geklickt hat, soll die Übersicht genau dieses Datensatzes gezeigt werden.

- Die betreffende ID wird übergeben
- Genau dieser Datensatz wird in der API SLIM gewählt und danach angezeigt
- Diese Anzeige liegt innerhalb eines Formulars, das wiederum dazu dient, die geänderten Daten dann mit dem Button „Änderung absenden“ in die Datenbank gesendet.



localhost/2020_rest2/termin_put.php?id=4

Ändern der Daten

Bitte ändern Sie die falschen Daten und drücken dann auf Änderungen absenden

ID
4

ID
4

Datum
2020-01-29

Inhalt
testdaten

Änderung absenden

Info:

- Einbinden der „httpful.phar“
- GET Import und Ablage des „values“ in eine neue Variable „\$id“. Bei der GET-Method wird der Name in eckigen Klammern und Anführungszeichen angeführt, die aus den Formularzeilen der Datei „termine_alle.php“ geholt wird (Zeile 43-45).
- Die „uri“ zeigt auf die API-SLIM „index“ im Ordner „public“ und übergibt die „\$id“.

```
16     require ("./httpful.phar");
17     $id = $_GET["id"]; //get Abfrage statt post
18
19     $uri = "http://localhost/2020_rest2/api/public/termin/$id";
20     $response = \Httpful\Request::get($uri)
21     ->expectsJson()
22     ->send();
```

Dann folgt das Öffnen des neuen Formulars, hier wieder mit der üblichen Methode POST. Dieses Formular übergibt an die neue Adresse „termin_put2.php“ die drei späteren Elemente und deren

„value“, die jeweils einen unterschiedlichen „name“ haben.

```
23
24     echo '<form action="termin_put2.php" method="post">';
25     echo '<br><h2>Ändern der Daten</h2>';
26     echo '<p>Bitte ändern Sie die falschen Daten und drücken dann auf
    Änderungen absenden</p>';
27
```

Darunter die eigentlichen Daten:

- Drei <input>-Felder mit jeweils unterschiedlichen „name“ und dahinter die „value“ mit den Elementen, die angezeigt werden, wie z.B. ID und Datum.

```
29
30     foreach($response -> body as $val)
31     {
32         echo '<p>ID<br><input class="form-control" readonly name="id"
    value="'. $id. '" ></p>';
33         echo '<p>ID<br><input class="form-control" readonly name="t_id" value="'.
    $val ->t_id. '"></p>';
34         echo '<p>Datum<br><input class="form-control" name="datum" value="'. $val ->
    datum. '"></p>';
35         echo '<p>Inhalt<br><input class="form-control" name="inhalt" value="'. $val
    -> inhalt. '"></p>';
36
37         echo '<button type="submit" class="btn btn-danger">Änderung
    absenden</button>';
38     }
39     echo "</form>";
40 }
```

```
41 |
42     ?>
43
44     </div> <!--ende container-->
45
46     <script src="js/jquery-3.3.1.min.js"></script>
47     <script src="js/bootstrap.js"></script>
48     </body>
49     </html>
50
```

Code:

```
<div class="container">
  <?php
    require ("./httpful.phar");
    $id = $_GET["id"]; //get Abfrage statt post
    $uri = "http://localhost/2020_rest2/api/public/termin/$id";
    $response = \Httpful\Request::get($uri)
    ->expectsJson()
    ->send();
```

```
echo '<form method="get" action="termin_put2.php">';
echo '<br><h2>Ändern der Daten</h2>';
echo '<p>Bitte ändern Sie die falschen Daten und drücken dann auf Änderungen absenden</p>';
```

```
foreach($response -> body as $val)
{
    echo '<p>ID<br><input class="form-control" readonly name="id" value="'. $id. '" ></p>';
    echo '<p>ID<br><input class="form-control" readonly name="t_id" value="'. $val ->t_id. '"></p>';
```

```

        echo'<p>Datum<br><input class="form-control" name="datum" value="'. $val ->
datum.'"></p>';
        echo'<p>Inhalt<br><input class="form-control" name="inhalt" value="'. $val ->
inhalt.'"></p>';

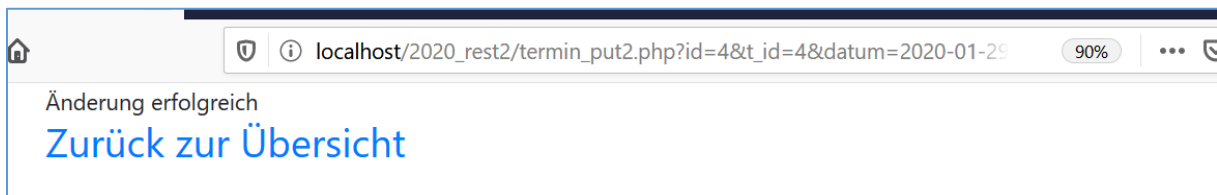
        echo'<button type="submit" class="btn btn-danger">Änderung absenden</button>';

echo"</form>";
}
?>

```

5b)termin_put2.php

Die Datei „termin_put2.php“ ändert dann entgültig mit der Methode „PUT“ den Datensatz.



```

11     </head>
12   <body>
13     <div class="container">
14       <?php
15         require ("./httpful.phar");
16         $id = $_GET["id"];
17
18         $uri = "http://localhost/2020_rest2/api/public/termin/update/$id";
19         $response = \Httpful\Request::put($uri)
20           ->sendsJson()
21           ->body ('{
22             "datum": "'. $_GET['datum'] .'",
23             "inhalt": "'. $_GET['inhalt'] .'"
24           }')
25           ->send();
26         echo "Änderung erfolgreich";
27         ?>
28         <br>
29         <h3><a href="termin_alle.php">Zurück zur Übersicht</a></h3>
30         <br>
31     </div>
32     <script src="js/jquery-3.3.1.min.js"></script>
33     <script src="js/bootstrap.js"></script>
34   </body>

```

6)Delete

6a)termin_delete.php

localhost/2020_rest2/termin_delete.php?id=5

Löschen der Daten

Wollen Sie wirklich diese Daten löschen, dann drücken dann auf den Button Löschen

ID

Datum

Inhalt

```
14 <body>
15 <div class="container">
16 <?php
17     session_start();
18     require ("./httpful.phar");
19     $id = $_GET["id"]; //get Abfrage statt post
20
21     $uri = "http://localhost/2020_rest2/api/public/termin/$id";
22     $response = \Httpful\Request::get($uri)
23     ->expectsJson()
24     ->send();
25
26     //normales Formular mit action und method
27     echo '<form method="get">';
28     echo '<br><h2>Löschen der Daten</h2>';
29     echo '<p>Wollen Sie wirklich diese Daten löschen, dann drücken dann auf den
30     Button Löschen</p>';
31
32     foreach($response -> body as $val)
33     {
34         echo '<p>ID<br><input class="form-control" name="tid" value="'. $val -> t_id
35         .'></p>';
36         echo '<p>Datum<br><input class="form-control" name="datum" value="'. $val ->
37         datum.'"></p>';
38         echo '<p>Inhalt<br><input class="form-control" name="inhalt" value="'. $val
39         -> inhalt.'"></p>';
40
41         echo '<button type="submit" class="btn btn-danger"
42         formaction="termin_delete2.php" name="id" value="'. $id.'" >Daten
43         löschen</button>|';
44
45     }
46     echo"</form>";
47 }
48 ?>
```

6b)termin delete2.php

```
1 <!DOCTYPE html>
2 <html lang="de">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
  fit=no">
6 <meta name="description" content="">
7 <meta name="author" content="">
8 <title>Termin anzeigen</title>
9 <!-- Bootstrap core CSS -->
10 <link href="css/bootstrap.css" rel="stylesheet">
11 </head>
12 <body>
13 <div class="container">
14 <?php
15 require ("./httpful.phar");
16 $id = $_GET["id"];
17
18 $uri = "http://localhost/2020_rest2/api/public/termin/delete/$id";
19 $response = \Httpful\Request::delete($uri)
20 ->sendJson()
21 ->body ('{ }')
22 ->send();
23 echo 'Löschung erfolgreich';
24 ?>
25 <br>
26 <h3><a href="termin_alle.php">Zurück zur Übersicht</a></h3>
27 <br>
28 </div>
29 <script src="js/jquery-3.3.1.min.js"></script>
30 <script src="js/bootstrap.js"></script>
31 </body>
32 </html>
```