

Registrieren und Login mit HTTPful und API (Slim3)

Öffne die Datei „registrieren.php“ und speichere sie unter „registrieren_api.php“.

Füge in der „action“ des Formulars den neuen Pfad ein, da jetzt die „api“-Version folgt.

```
30
31 ▾ <form action="includes/registrieren_insert_api.php"
    method="post" enctype="multipart/form-data">
32 ▾ <div class="form-group">
```

Lösche das Input-Feld bezüglich Bild. Hier soll es nicht verwendet werden.

```
46 </div>
47 ▾ <div class="form-group">
48 <input type="file" name="bild" class="form-control"
    enctype="multipart/form-data">
49 </div>
50 <br>
```

Und dazu passend auch im Formulkopf den „enctype“-Bereich, weil er somit nicht benötigt wird.

```
31 ▾ <form action="includes/registrieren_insert_api.php" method="post"
    enctype="multipart/form-data">
32 ▾ <div class="form-group">
```

Kopiere danach im Ordner „includes“ die Datei, welche die Registrierung entgegennimmt:

prozess.php	07.0
prozess_api.php	16.0
registrieren_insert - Kopie.php	07.0
registrieren_insert.php	07.0

Speicher die Kopie als „registrieren_insert_api.php“ und öffne diese für die ersten Änderungen.

registrieren insert api.php

- Lösche aus Zeile 12 die Variablenerstellung bezüglich Bild.

```
10 $klasse = $_POST["klasse"];
11 $geb = $_POST["geb"];
12 $bild = addslashes(file_get_contents($_FILES['bild']['tmp_name']));
13 $benutzername = $_POST["benutzername"];
14 $email = $_POST["email"];
```

- Entferne die alte „require Verbindung“ und ersetze diese durch die neue. Auch hier muss man auf die beiden Punkte achten.

```
15 $pass = $_POST["pass#"];
16 |
17 require ('../httpful.phar'); //für localhost
18 //require ("/usr/share/httpful/bootstrap.php"); für extern
19
20 if(empty($benutzername) || empty($email) || empty($pass) )
```

Danach entferne die SQL-INERT Variante und ersetze es durch HTTPful-Code

```
26 ▼ else {
27     $hashedPwd = password_hash($pass, PASSWORD_DEFAULT);
28
29 ▼     $sql = "INSERT INTO schueler (vorname, nachname, geschlecht,
        klasse, geb, bild, benutzername, email, pass) VALUES
        ('$vorname', '$nachname', '$geschlecht', '$klasse', '$geb',
        '$bild', '$benutzername', '$email', '$hashedPwd)";
30
31     $erg = $con->query ($sql) or die("Fehler bei der
        Speicherung.");
32
33     header("Location: ../registrieren.php?error=<div class='alert
        alert-success'>Sie sind erfolgreich angemeldet</div>");
34     exit();
```

Zuerst den Pfad der URI auf den Pfad in der API:

```
27     $hashedPwd = password_hash($pass, PASSWORD_DEFAULT);
28
29     $uri =
        "http://localhost/2020_3ck_db_ueben/api/public/schueler/registrieren";|
30
31     header("Location: ../registrieren.php?error=<div class='alert alert-
```

Öffne danach die „aendern_insert_api.php“, da dort Code zum Kopieren liegt. ☺

Kopiere von dort:

```
13     $response = \Httpful\Request::put($uri)
14     ->sendJson()
15     ->body ('{
16         "vorname": "'. $_POST['vorname'].'",
17         "nachname": "'. $_POST['nachname'].'",
18         "geschlecht": "'. $_POST['geschlecht'].'",
19         "klasse": "'. $_POST['klasse'].'",
20         "geb": "'. $_POST['geb'].'",
21         "benutzername": "'. $_POST['benutzername'].'",
22         "email": "'. $_POST['email'].'"
23     }')
24     ->send();
```

Ändere daran einiges:

- Zuerst handelt es sich hier nicht um die Methode „put“ sondern „post“, wir wollen ja nichts ändern, sondern komplett neu in die Datenbank schreiben.
- Im „body“ verwende nach den korrekten Daten, die mit der Datenbank übereinstimmen müssen (vor dem :) nun nicht die \$_POST-Variablen, sondern, da sie ja oben ab Zeile 7 vorhanden sind, die neuen Variablen. Diese sind innerhalb der Verknüpfungspunkte einzufügen.
 - Ein Vorteil der weiteren Verwendung der Variablen ist, dass die IF in Zeile 20 auch mit den Variablennamen weiterhin arbeiten kann.
- Das Passwort fehlt aus der Kopie, daher muss es hier dazu geschrieben werden. Das kann perfekt genutzt werden, um auch des gehashte Passwort aus der Zeile 27 mit zu berücksichtigen.

```

30
31     $response = \Httpful\Request::post($uri)
32     ->sendJson()
33     ->body ('{
34         "vorname": "'. $vorname .'",
35         "nachname": "'. $nachname .'",
36         "geschlecht": "'. $geschlecht .'",
37         "klasse": "'. $klasse .'",
38         "geb": "'. $geb .'",
39         "benutzername": "'. $benutzername .'",
40         "email": "'. $email .'",
41         "pass": "'. $hashedPwd .'"
42     }')
43     ->send();

```

Code:

```

$response = \Httpful\Request::post($uri)
->sendJson()
->body ('{
    "vorname": "'. $vorname .'",
    "nachname": "'. $nachname .'",
    "geschlecht": "'. $geschlecht .'",
    "klasse": "'. $klasse .'",
    "geb": "'. $geb .'",
    "benutzername": "'. $benutzername .'",
    "email": "'. $email .'",
    "pass": "'. $hashedPwd .'"
}')
->send();

```

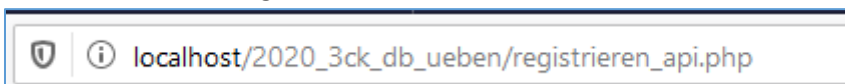
Zum Vergleich die Variablen, die hier eingebaut wurden:

```

6 //hier werden Variablen erstellt, die aus dem Formular kommen
7 $vorname = $_POST['vorname'];
8 $nachname = $_POST["nachname"];
9 $geschlecht = $_POST["geschlecht"];
10 $klasse = $_POST["klasse"];
11 $geb = $_POST["geb"];
12
13 $benutzername = $_POST["benutzername"];
14 $email = $_POST["email"];
15 $pass = $_POST["pass#"];

```

Testen. Beachte weiterhin, dass beim Testen man aus der „api“-Version heraus testet. Das muss händisch berücksichtigt werden:



Login

Kopiere im Ordner „includes“ die Datei „login.php“ und benenne die Kopie „login_api.php“.

Öffne diese für die ersten Veränderungen.

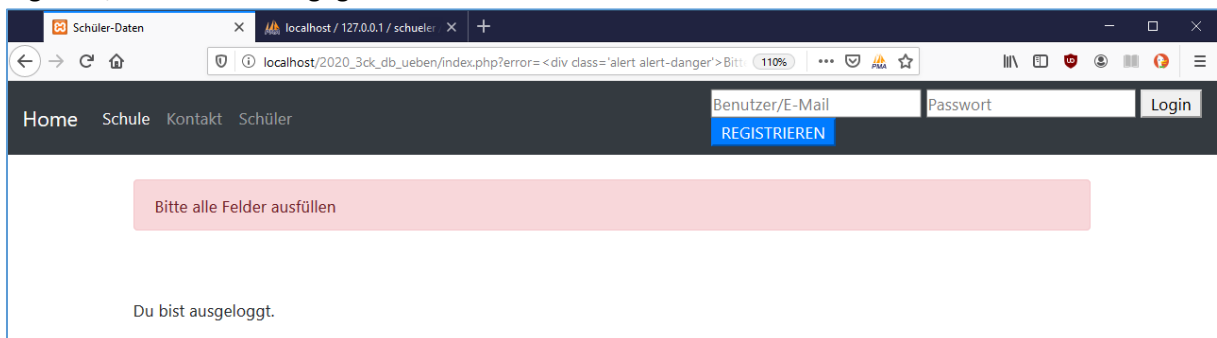
Entferne die „Verbindung“ und kopiere die neue „httpful“ stattdessen hinein. Zeile 7+8.

```
1 <?php
2
3 //wenn Button geklickt
4 if (isset($_POST['login-submit']))
5 {
6
7 require ('../httpful.phar'); //für localhost
8 //require ("usr/share/httpful/bootstrap.php"); für extern
9
10 $login_user = $_POST['mailuid'];
```

Darunter bleibt ebenfalls alles beim Alten, bis zur „else“. Es werden nämlich weiterhin zwei Variablen erzeugt und anhand dieser überprüft, ob die Input-Felder ausgefüllt wurden. Die Fehlermeldung bleibt bestehen.

```
11 $login_user = $_POST['mailuid'];
12 $login_pwd = $_POST['pwd'];
13
14 if(empty($login_user) || empty($login_pwd) )
15 {
16     header("Location: ../index.php?error=<div class='alert alert-
17     danger'>Bitte alle Felder ausfüllen</div>");
18     exit();
19 }
19 else {
```

Ergebnis, wenn nichts eingegebene wird:



In der „else“-Verzweigung wird nun aber

```
19 else {
20     $uri = "http://localhost/2020_3ck_db_ueben/api/public/schueler/login";
21     $response = \Httpful\Request::get($uri)
22         ->expectsJson()
23         ->send();
```

```
$uri = "http://localhost/2020_3ck_db_ueben/api/public/schueler/login";
$response = \Httpful\Request::get($uri)
```

```
->expectsJson()  
->send();
```

Danach ist der komplette Code der die Daten aus Datenbank und aus der Eingabe durch den User im „header“ in das API verschwunden. Dort wird nun der Vergleich angestellt.

Das ist ein Vorteil, da hier der Code sich nur mehr auf die Behandlung der Ausgabe bei den Fehlermeldungen konzentriert.

```
24  
25 ▼     if ($message == 'falsches Passwort') {  
26         header("Location: ../index.php?error=<div class='alert alert-  
27             danger'>Passwort falsch</div>");  
28         exit();  
29 ▼     }  
30     else if ($message == 'success') {  
31         header("Location: ../adminbereich.php?error=<div  
32             class='alert alert-success'>Willkommen. </div>");  
33         exit();  
34     }  
35     else {  
36         header("Location: ../index.php?error=<div class='alert alert-  
37             danger'>Kein User/E-Mail gefunden </div>");  
38         exit();  
39     }  
40 }>
```

Code:

```
if ($message == 'falsches Passwort') {  
    header("Location: ../index.php?error=<div class='alert alert-danger'>Passwort falsch</div>");  
    exit();  
}  
else if ($message == 'success') {  
    header("Location: ../adminbereich.php?error=<div class='alert alert-success'>Willkommen.  
</div>");  
    exit();  
}  
else {  
    header("Location: ../index.php?error=<div class='alert alert-danger'>Kein User/E-Mail  
gefunden </div>");  
    exit();  
}
```

Leider ist ein grüner Erfolgsbutton in der Datei „adminbereich.php“ nicht installiert, obwohl die Erfolgsmeldung „Willkommen“ vorbereitet wäre. Aber das muss ja nicht sein, weil man sowieso erfolgreich in den verborgenen Bereich eingelassen wurde und somit sein Ziel erreicht hat.

Ergebnis, wenn das Passwort falsch eingegeben wird:

