

Ionic – Daten an Server senden

Inhalt:

1. Daten an Server senden und zurückerhalten mit api.php (kein Speichern)
2. Daten in eine Localhost-Datenbank schreiben
3. auf externen Server „Domanfactory“ testen
4. Erweiterung zu Dateneingabe mit „username“ und „password“ (localhost)

1) Daten an Server senden und zurückerhalten mit PHP-Datei

Erstelle ein neues Projekt namens „form“ mit der Eigenschaft „blank“.

```
C:\Users\edi\Desktop\form1>ionic start form blank
```

home.html

Es ist hier keine <form> aber mit Hilfe des „input-Feldes“ gilt es als solches.

In Zeile 13 wird eine „Two-Way-Binding“ verwendet, nämlich durch [(ngModel)]="". Es sorgt dafür, dass bei einer Änderung des Eingabeelements der veränderte Status sofort in der Komponente zur Verfügung steht. Dafür muss das „ngModel“ zwischen runden und eckigen Klammern stehen. Wir verwenden die Eingabe dann in der Funktion „submit“.

Der Button unten, Zeile 16, hat eine „click-Funktion“, die hier „submit“ heißt, aber auch anders heißen könnte.

Diese Funktion wird ausgeführt, wenn man darauf klickt. Dafür wird diese in „home.ts“ erstellt.

```
9 <ion-content padding>
10 <ion-list>
11 <ion-item>
12 <ion-label floating>Username</ion-label>
13 <ion-input type="text" name="username" [(ngModel)]="data.username"></ion-input>
14 </ion-item>
15
16 <button ion-button block (click)="submit()">Submit to server</button>
17 </ion-list>
18
19 <ion-card>
20 <ion-card-header>
21 | Response
22 </ion-card-header>
23
24 <ion-card-content>
25 | <b>{{data.response}}</b>
26 </ion-card-content>
27 </ion-card>
28 </ion-content>
```

home.ts

Zeile 10 einfügen

```
8 export class HomePage {
9
10 data:any = {};
11
```

Die public http und darunter dreimal this

```

13 constructor(public navCtrl: NavController, public http: Http) {
14     this.data.username = '';
15     this.data.response = '';
16     this.http = http;
17 }

```

Ganz oben wahrscheinlich automatisch – Zeile 3

```

<> home.html x TS home.ts x
1 import { Component } from '@angular/core';
2 import { NavController } from 'ionic-angular';
3 import { Http } from '@angular/http';
4
5 @Component({

```

Der Link in Zeile 20 führt zum Localhost-Ordner von Xampp. Dort liegt die Datei „api.php“ im Ordner „form“:

```

19 submit() {
20     var link = 'http://localhost/form/api.php';
21     var myData = JSON.stringify({username: this.data.username});
22
23     this.http.post(link, myData)
24     .subscribe(data => {
25         this.data.response = data["_body"];
26     }, error => {
27         console.log("Oooops!");
28     });
29 }

```

app.module.ts:

Importiere das HttpModule in Zeile 10 aber auch in 20.

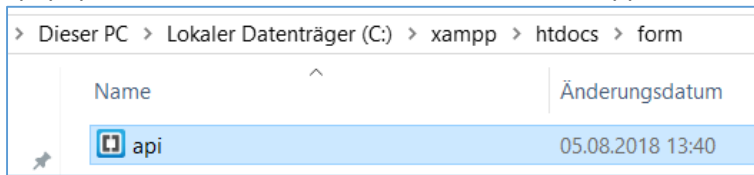
```

<> home.html TS home.ts TS app.module.ts ●
1 import { BrowserModule } from '@angular/platform-browser';
2 import { ErrorHandler, NgModule } from '@angular/core';
3 import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';
4 import { SplashScreen } from '@ionic-native/splash-screen';
5 import { StatusBar } from '@ionic-native/status-bar';
6
7 import { MyApp } from './app.component';
8 import { HomePage } from '../pages/home/home';
9
10 import { HttpModule } from '@angular/http';
11
12 @NgModule({
13     declarations: [
14         MyApp,
15         HomePage
16     ],
17     imports: [
18         BrowserModule,
19         IonicModule.forRoot(MyApp),
20         HttpModule
21     ],
22     bootstrap: [IonicApp],

```

api.php auf den Server legen:

api.php in htdocs und dem Ordner „form“ von xampp:



```
<?php
```

```
//http://stackoverflow.com/questions/18382740/cors-not-working-php
if (isset($_SERVER['HTTP_ORIGIN'])) {
    header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}");
    header('Access-Control-Allow-Credentials: true');
    header('Access-Control-Max-Age: 86400'); // cache for 1 day
}

// Access-Control headers are received during OPTIONS requests
if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS') {

    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))
        header("Access-Control-Allow-Methods: GET, POST, OPTIONS");

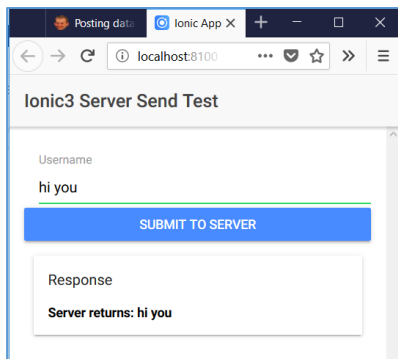
    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))
        header("Access-Control-Allow-Headers:
{$_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");

    exit(0);
}

//http://stackoverflow.com/questions/15485354/angular-http-post-to-php-and-undefined
$postdata = file_get_contents("php://input");
if (isset($postdata)) {
    $request = json_decode($postdata);
    $username = $request->username;

    if ($username != "") {
        echo "Server returns: " . $username;
    }
    else {
        echo "Empty username parameter!";
    }
}
else {
    echo "Not called properly with username parameter!";
}
?>
```

Ergebnis:



Quelle:

<http://www.nikola-breznjak.com/blog/javascript/ionic3/posting-data-ionic-3-app-php-server/>

Auf Github:

<https://github.com/Hitman666/Ionic3ServerSendTest>

2) Daten in eine Localhost-Datenbank schreiben

Erstelle eine Datenbank namens „ioniclogin“ mit der Tabelle „form“:



Erstelle aus der Datei „api.php“ eine „input.php“.

Diese soll nun nicht die Daten zurück in den Webbrowser geben, sondern in die Datenbank schreiben mit Hilfe von „INSERT INTO“:

Belasse den oberen Teil gleich und entferne die „IF-Abfrage“.

Erstelle dafür aber die Verbindung zur Datenbank und die „INSERT INTO“ Abfrage.

```
20
21 $postdata = file_get_contents("php://input");
22     $request = json_decode($postdata);
23     $username = $request->username;
24
25 /* Verbindung aufnehmen*/
26 $con = new MySQLi("localhost", "root", "", "ioniclogin");
27 if ($con->connect_error) {
28     echo "Fehler bei der Verbindung: " . mysqli_connect_error();
29     exit();
30 }
31
32 $sql = "INSERT INTO form (username) VALUES ('$username')";
33
34 $erg = mysqli_query($con, $sql)
35     or die("leider Fehler");
36 |
37 ?>
```

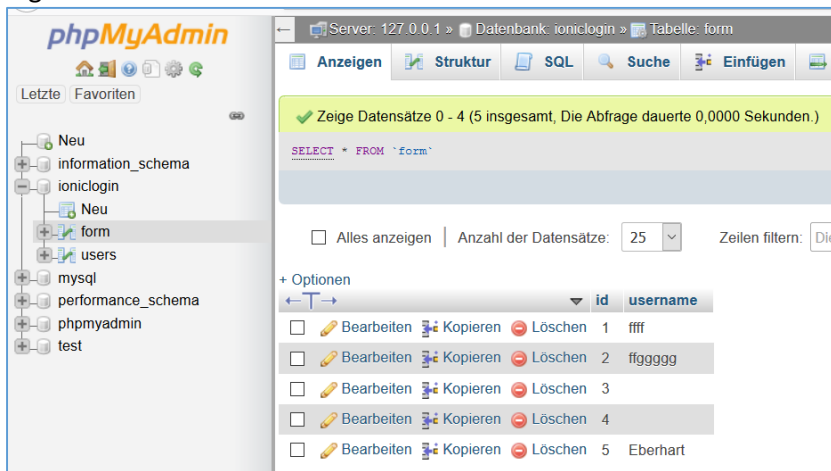
Umstellung auf die neue Datei am Server in „home.ts“ lautend auf „input.php“:

```

19 submit() {
20   var link = 'http://localhost/form/input.php';
21   var myData = JSON.stringify({username: this.data.username});
22
23   this.http.post(link, myData)
24     .subscribe(data => {
25       this.data.response = data["_body"];
26     }, error => {
27       console.log("Oooops!");
28     });
29 }

```

Ergebnis:



Die „id 3 und 4“ waren Fehlversuche, die aber auch die ID verbraucht haben.

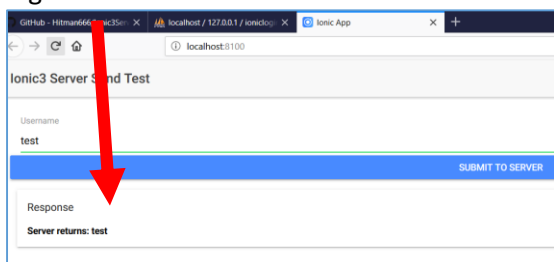
Um auch eine Ausgabe zu erhalten kann man aber Teil aus „api.php“ in „input.php“, nach der Datenbank-Abfrage wiederverwenden:

```

32 $sql = "INSERT INTO form (username) VALUES ('$username')";
33
34 $erg = mysqli_query($con, $sql)
35   or die("leider Fehler");
36
37 if ($username != "") {
38     echo "Server returns: " . $username;
39 }
40 else {
41     echo "Empty username parameter!";
42 }
43
44 ?>

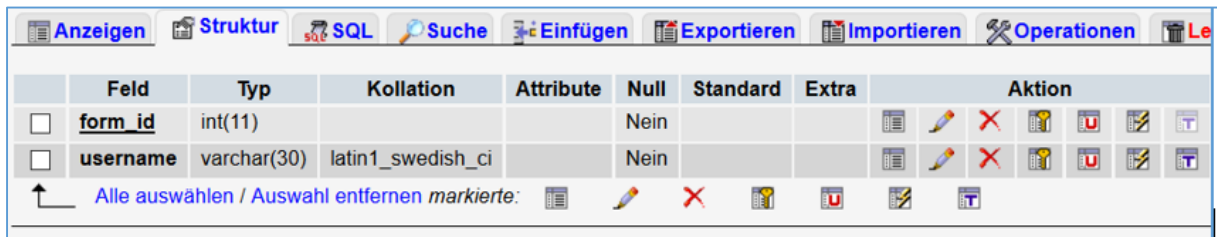
```

Ergebnis:



3) auf Server „Domanfactory“ testen

Tabelle „form“ anlegen:



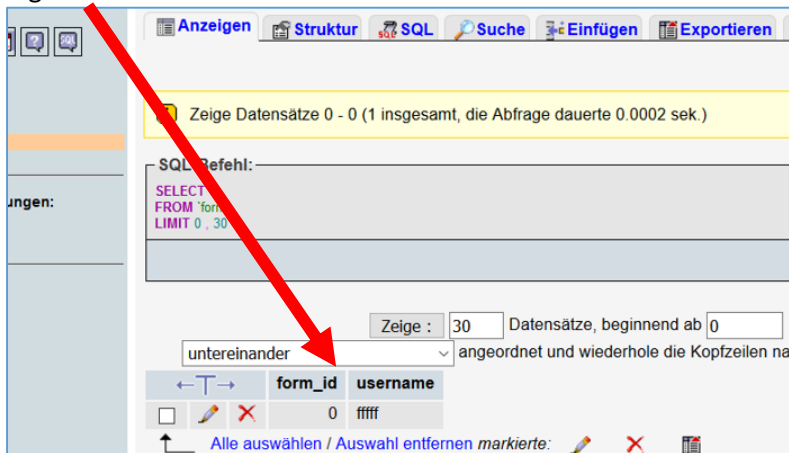
	Feld	Typ	Kollation	Attribute	Null	Standard	Extra	Aktion
<input type="checkbox"/>	form_id	int(11)			Nein			
<input type="checkbox"/>	username	varchar(30)	latin1_swedish_ci		Nein			

Alle auswählen / Auswahl entfernen markierte:

Ändere die Datenbank-Verbindung:

```
21 $postdata = file_get_contents("php://input");
22     $request = json_decode($postdata);
23     $username = $request->username;
24
25 /* Verbindung aufnehmen*/
26 $con = new MySQLi("mysql5.eberhart.media", " Benutzer " " passwort .", Datenbank );
27 if ($con->connect_error) {
28     echo "Fehler bei der Verbindung: " . mysqli_connect_error();
29     exit();
30 }
```

Ergebnis:



Zeige Datensätze 0 - 0 (1 insgesamt, die Abfrage dauerte 0.0002 sek.)

SQL Befehl:

```
SELECT
FROM form
LIMIT 0, 30
```

Zeige : 30 Datensätze, beginnend ab 0

untereinander angeordnet und wiederhole die Kopfzeilen na

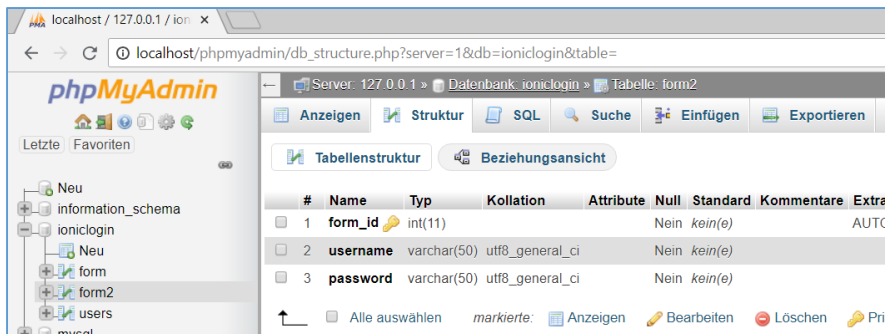
form_id	username
0	ffff

Alle auswählen / Auswahl entfernen markierte:

4) Erweiterung zu Dateneingabe mit „username“ und „password“

Speichern in Localhost-Datenbank

Neue Tabelle in Datenbank mit „username“ und „password“. Name der Tabelle: form2



Anpassungen an die neue Tabelle

In Xampp die „input2.php“:

- Auch postdata und request für „password“
- Zeile 36 in Insert aufnehmen
- Ausgabe verlängern in Zeile 42.

```
21 $postdata = file_get_contents("php://input");
22     $request = json_decode($postdata);
23     $username = $request->username;
24
25 $postdata = file_get_contents("php://input");
26     $request2 = json_decode($postdata);
27     $password = $request2->password;
28
29 /* Verbindung aufnehmen*/
30 $con = new MySQLi("localhost", "root", "", "ioniclogin");
31 ▼ if ($con->connect_error) {
32     echo "Fehler bei der Verbindung: " . mysqli_connect_error();
33     exit();
34 }
35
36 $sql = "INSERT INTO form2 (username, password) VALUES ('$username', '$password')";
37
38 $erg = mysqli_query($con, $sql)
39     or die("leider Fehler");
40
41 ▼ if ($username != "") {
42     echo "Server returns: " . $username . " und Passwort" . $password;
43 }
```

Änderung in „home.html“:

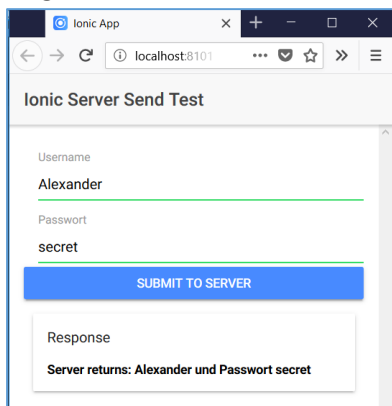
```
10 <ion-list>
11   <ion-item>
12     <ion-label floating>Username</ion-label>
13     <ion-input type="text" name="username" [(ngModel)]="data.username">/ion-input>
14   </ion-item>
15   <ion-item>
16     <ion-label floating>Passwort</ion-label>
17     <ion-input type="text" name="password" [(ngModel)]="data.password">/ion-input>
18   </ion-item>
19
20   <button ion-button block (click)="submit()">Submit to server</button>
21 </ion-list>
```

home.ts

- Zeile 15 für this.data.password
- JSON.stringify mit password verlängern (Zeile 22)

```
13 constructor(public navCtrl: NavController, public http: Http) {
14   this.data.username = '';
15   this.data.password = '';
16   this.data.response = '';
17   this.http = http;
18 }
19
20 submit() {
21   var link = 'http://localhost/form/input2.php';
22   var myData = JSON.stringify({username: this.data.username, password: this.data.password});
23 }
```

Eingabe:



Ergebnis:

ID 3 ist schon super: 1 und 2 waren noch fehlerhaft, da ohne Passwort von mir eingeben.

