

API, Testen mit postman

Inhalt

1. Die API-Schnittstelle (SLIM 3) index.php
2. Datenbank-Daten in db.php
3. API auf dem Server
4. Zugriff per Postman

Es wird die Datenbank auf dem Server hosttech.at genutzt. Der Nutzer dort ist ioniceinkauf.digbizmistelbach.info.

Datenbanken für ioniceinkauf.digbizmistelbach.info ▾

Folgende Tabellen sind dort vorhanden:

	Tabelle	Aktion
<input type="checkbox"/>	schueler	★ Anzeigen
<input type="checkbox"/>	schuhe	★ Anzeigen
<input type="checkbox"/>	user	★ Anzeigen
3 Tabellen		Gesamt

schuhe:

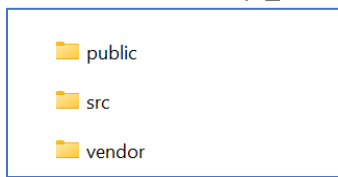
id	name	preis	rating	beschreibung	bildurl
1	Delta run	240	4.5	Keilsandalette, rainbow	https://ioniceinkauf.digbizmistelbach.info/urlimg/...
2	Delta	280	4.7	Plateausandalette, rainbow	https://ioniceinkauf.digbizmistelbach.info/urlimg/...
3	Ko Mid	200	4.2	Keilsandalette, malibu	https://ioniceinkauf.digbizmistelbach.info/urlimg/...
4	Rico	240	4.8	Plateausandalette, summer	https://ioniceinkauf.digbizmistelbach.info/urlimg/...
5	Strappy Hi	180	4.5	Hi Heal Sandalette	https://ioniceinkauf.digbizmistelbach.info/urlimg/...
6	Delta	280	4.6	Plateausandalette, bronze	https://ioniceinkauf.digbizmistelbach.info/urlimg/...

user: Achtung die ID lautet „iduser“

iduser	mail	benutzername	passwort
1	vanessa@gmail.com	vanessa	62c3673ffdc779eb08e1f3ae0ff40996
4	test@gmail.com	test	1234
5	test1@gmail.com	test2	81dc9bdb52d04dc20036dbd8313ed055
6	hallo@gmail.com	hallo	Franz
7	vanessa@hak.at	vani	81dc9bdb52d04dc20036dbd8313ed055

1) Die API-Schnittstelle (SLIM 3) index.php

Öffne den Ordner „api_erste“ und darin der Ordner „public“ und mit dem Editor index.php.



Diese „index.php“ bezieht sich auf die Datei, welche den Zugang zur Datenbank trifft. Das ist die „db.php“ im Ordner „config“:

```
14 // # include DB connection file
15 require '../src/config/db.php';
16
17 $app = new \Slim\App;
```

In der Zeile 17 sieht man auch den Start der SLIM-API die UNBEDINGT in der letzten Zeile der gesamten „index.php“ enden muss (diese Zeile darf niemals fehlen):

```
270
271 $app->run();
```

Zwischen diesen Zeilen kann man unendlich viele Abfragen erstellen.

Die erste hier ist eine normale GET-Abfrage in typischer SQL/PHP-Codierung:

```
17 $app = new \Slim\App;
18
19 $app->get('/schuhe_holen', function( Request $request, Response $response){
20
21     $sql = "SELECT id,name,preis,rating,beschreibung FROM schuhe";
22
23     try {
24         $db = new db();
25         $db = $db->connect();
26         $stmt = $db->query( $sql );
27         $schuhe = $stmt->fetchAll( PDO::FETCH_OBJ );
28         $db = null; // clear db object
```

```
29
30         echo json_encode( $schuhe );
31     } catch( PDOException $e ) {
32         // show error message as json format
33         echo '{"error": {"msg": ' . $e->getMessage() . '}}';
34     }
35 }
36
37 });
```

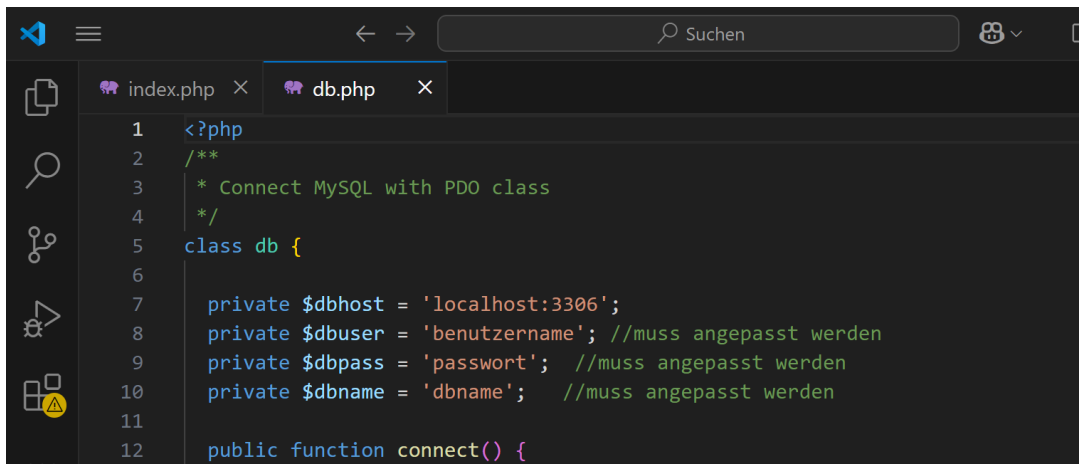
Zeile 27 und 30: der Name der Variable \$schuhe ist frei wählbar, einzig muss gewährleistet sein, dass die beiden gleich sind.

Zeile 21: der Name des Schuheres ist in der Datenbank „name“. In SQL ist das aber ein vordefinierter Begriff (wie auch password). Daher wird er hier in blauer Farbe dargestellt. Das könnte zu Problemen führen. Daher in Zukunft bitte einen anderen Begriff wählen z.B. „bezeichnung“ oder auch „namen“.

In Zeile 37 wird diese GET-Abfrage wieder geschlossen.

2) Datenbank-Daten in db.php

Die Details für den Zugriff zur Datenbank liegen in „src/config/db.php“.



```
1 <?php
2 /**
3  * Connect MySQL with PDO class
4  */
5 class db {
6
7     private $dbhost = 'localhost:3306';
8     private $dbuser = 'benutzername'; //muss angepasst werden
9     private $dbpass = 'password'; //muss angepasst werden
10    private $dbname = 'dbname'; //muss angepasst werden
11
12    public function connect() {
```

Host	localhost:3306
Datenbankname	apitest
Benutzername	apitest_digbiz
Passwort	*****

PSW: BITTE beim IM-Lehrer nachfragen (Beachte auch den Punkt hinten)

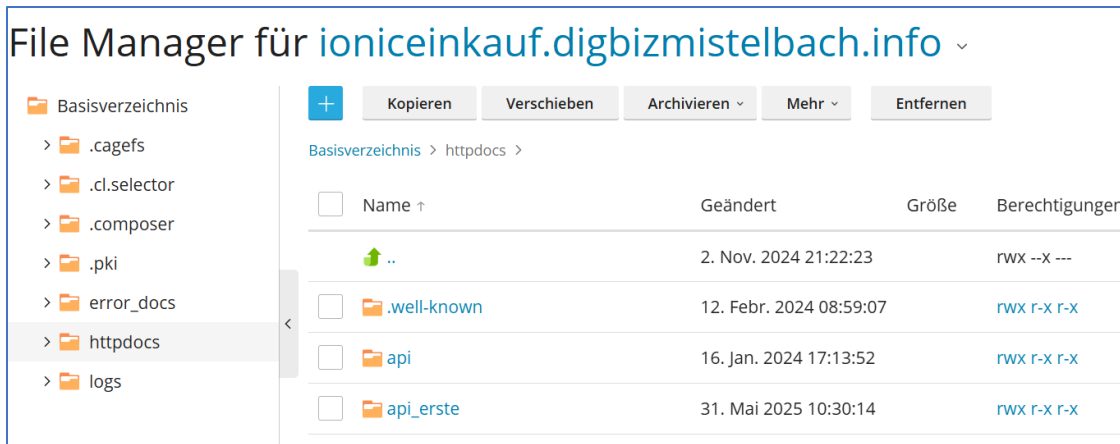
Ergebnis:



```
7     private $dbhost = 'localhost:3306';
8     private $dbuser = 'apitest_digbiz'; //muss angepasst werden
9     private $dbpass = '          ' ;s angepasst werden
10    private $dbname = 'apitest'; //muss angepasst werden
```

3) API auf dem Server

Dieses API liegt bereits auf dem Server, damit alle zugreifen können.



4) Zugriff per Postman

Öffne die App postman



Nun sollen ein paar Infos abgeholt werden.

Wir haben folgendes bereits in der „index.php“ vorbereitet:

```

19 $app->get('/schuhe_holen', function( Request $request, Response $response){
20
21     $sql = "SELECT id,name,preis,rating,beschreibung FROM schuhe";
22

```

```

42 $app->get('/schuhe_nr5', function( Request $request, Response $response){
43     // echo 'Benutzer';
44
45     $sql = "SELECT id,name,preis,rating,beschreibung FROM kunden WHERE id = 5";
46

```

```

177 // alle user
178 $app->get('/user_holen', function( Request $request, Response $response){
179     // echo 'Benutzer';
180
181     $sql = "SELECT * FROM user";
182

```

```

200 // mit ausgesuchter ID
201 $app->get('/user/{id}', function( Request $request, Response $response){
202     $id = $request->getAttribute('id');
203     $sql = "SELECT * FROM user WHERE id = '$id' ";
204

```

Eine POST-Abfrage für login

```

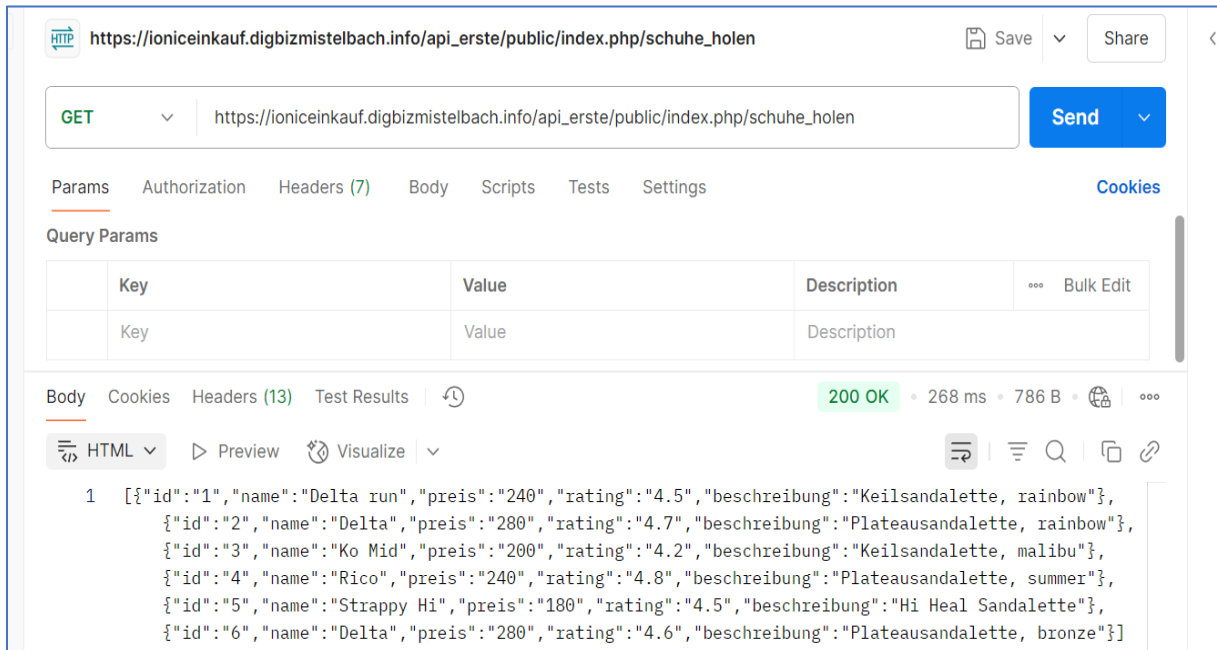
120 $app->post('/login', function( Request $request, Response $response){
121
122     $email = $request->getParam('email'); //in Klammer email von der log
123     $password = $request->getParam('password'); //in Klammer password von

```

Nutze Postman:

In die Zeile eingeben:

https://ioniceinkauf.digbizmistelbach.info/api_erste/public/index.php/schuhe_holen



https://ioniceinkauf.digbizmistelbach.info/api_erste/public/index.php/schuhe_holen

GET https://ioniceinkauf.digbizmistelbach.info/api_erste/public/index.php/schuhe_holen

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

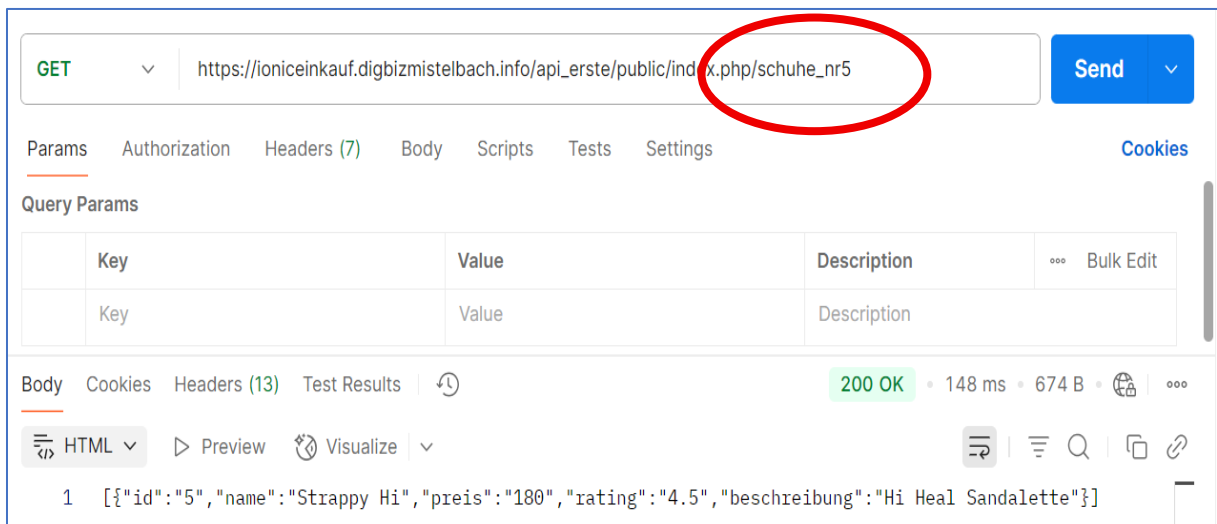
Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (13) Test Results 200 OK • 268 ms • 786 B

```
1 [{"id": "1", "name": "Delta run", "preis": "240", "rating": "4.5", "beschreibung": "Keilsandalette, rainbow"}, {"id": "2", "name": "Delta", "preis": "280", "rating": "4.7", "beschreibung": "Plateausandalette, rainbow"}, {"id": "3", "name": "Ko Mid", "preis": "200", "rating": "4.2", "beschreibung": "Keilsandalette, malibu"}, {"id": "4", "name": "Rico", "preis": "240", "rating": "4.8", "beschreibung": "Plateausandalette, summer"}, {"id": "5", "name": "Strappy Hi", "preis": "180", "rating": "4.5", "beschreibung": "Hi Heal Sandalette"}, {"id": "6", "name": "Delta", "preis": "280", "rating": "4.6", "beschreibung": "Plateausandalette, bronze"}]
```

Ändere den Endpunkt auf: schuhe_nr5



GET https://ioniceinkauf.digbizmistelbach.info/api_erste/public/index.php/schuhe_nr5

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (13) Test Results 200 OK • 148 ms • 674 B

```
1 [{"id": "5", "name": "Strappy Hi", "preis": "180", "rating": "4.5", "beschreibung": "Hi Heal Sandalette"}]
```

Ändere den Endpunkt auf: user_holen

GET `https://ioniceinkauf.digbizmistelbach.info/api_erste/public/index.php/user_holen` Send

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (13) Test Results 200 OK • 80 ms • 813 B

HTML Preview Visualize

```
1 [{"iduser": "1", "mail": "vanessa@gmail.com", "benutzername": "vanessa",
  "passwort": "62c3673ffdc779eb08e1f3ae0ff40996"}, {"iduser": "4", "mail": "test@gmail.com",
  "benutzername": "test", "passwort": "1234"}, {"iduser": "5", "mail": "test1@gmail.com",
  "benutzername": "test2", "passwort": "81dc9bdb52d04dc20036dbd8313ed055"}, {"iduser": "6",
  "mail": "hallo@gmail.com", "benutzername": "hallo", "passwort": "Franz"}, {"iduser": "7",
  "mail": "vanessa@hak.at", "benutzername": "vani", "passwort": "81dc9bdb52d04dc20036dbd8313ed055"},
  {"iduser": "8", "mail": "hallo@21", "benutzername": "daniel",
  "passwort": "81dc9bdb52d04dc20036dbd8313ed055"}, {"iduser": "10", "mail": "test@gmail.com",
  "benutzername": "test1", "passwort": "1234"}, {"iduser": "11", "mail": "vanessa@hak.at",
  "benutzername": "theresa", "passwort": "81dc9bdb52d04dc20036dbd8313ed055"}]
```

Diese Abfrage ist sehr flexibel, da die ID wählbar ist:

GET `https://ioniceinkauf.digbizmistelbach.info/api_erste/public/index.php/user/1` Send

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (13) Test Results 200 OK • 132 ms • 688 B

HTML Preview Visualize

```
1 [{"iduser": "1", "mail": "vanessa@gmail.com", "benutzername": "vanessa",
  "passwort": "62c3673ffdc779eb08e1f3ae0ff40996"}]
```

iduser	mail	benutzername	passwort
1	vanessa@gmail.com	vanessa	62c3673ffdc779eb08e1f3ae0ff40996
4	test@gmail.com	test	1234
5	test1@gmail.com	test2	81dc9bdb52d04dc20036dbd8313ed055
6	hallo@gmail.com	hallo	Franz
7	vanessa@hak.at	vani	81dc9bdb52d04dc20036dbd8313ed055

LOGIN: ist eine POST keine GET

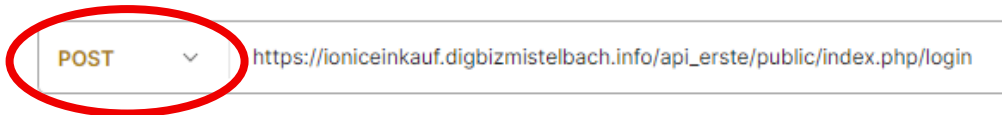
4 test@gmail.com test 1234

```
120 $app->post('/login', function( Request $request, Response $response){
121
122     $email = $request->getParam('email'); //in Klammer email von der logi
123     $passwort = $request->getParam('passwort'); //in Klammer password von d
```

Eingeben muss man die Parameter email und passwort

Vorgangsweise:

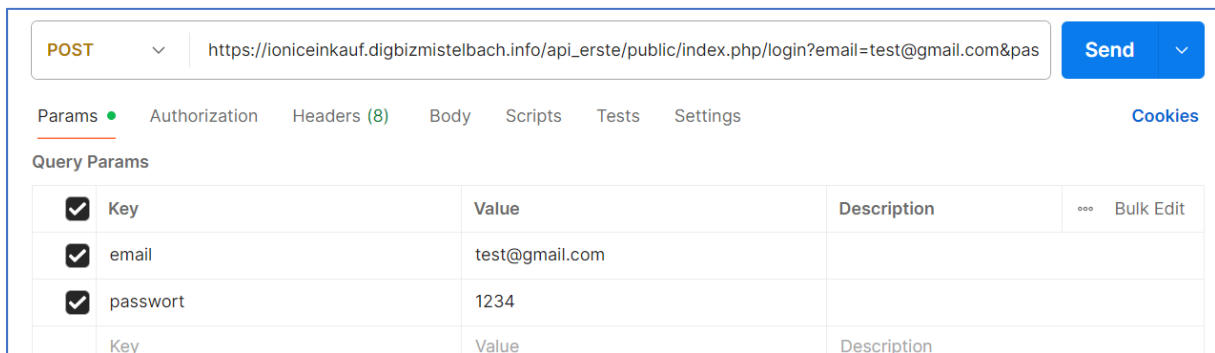
Zuerst umstellen auf POST, dann den Endpunkt auf „login“



https://ioniceinkauf.digbizmistelbach.info/api_erste/public/index.php/login

Da man aber die E-Mail und das Passwort im üblichen HTML-Formular in die Inputfelder eingeben muss, müssen diese beiden Parameter auch hier eingegeben werden:

Bei der Eingabe werden die Daten sofort in die URL mitübernommen:



Ergebnis:

