

SLIM 3 - Get und POST

1)SLIM installieren

Erstelle einen Ordner in XAMPP / htdocs namens „slim3“.

Genauere Nacharbeit von folgender perfekter Website:

<https://applerinquest.com/how-to-create-the-rest-api-with-slim-framework/>

Navigiere in der CMD in diesen neuen Ordern mittels den Befehlen

- cd..
- cd slim3

Navigiere zuerst auf C: Dann gibst den kompletten neuen Pfad ein. Beachte die Back-Slash. Wenn ein Anfangsbuchstabe eingegeben ist, kann man mit der gedrückten „TAB“-Taste den Autovervollständiger nutzen.

```

c:\> Eingabeaufforderung
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\edi>cd..

C:\Users>cd..

C:\>cd xampp

C:\xampp>cd htdocs

C:\xampp\htdocs>cd slim3

C:\xampp\htdocs\slim3>composer require slim/slim:3.*
```

Der Installationscode findet man auf der Slim-Homepage:



Installation

Upgrade Guide

Web Servers

Deployment

Tutorial

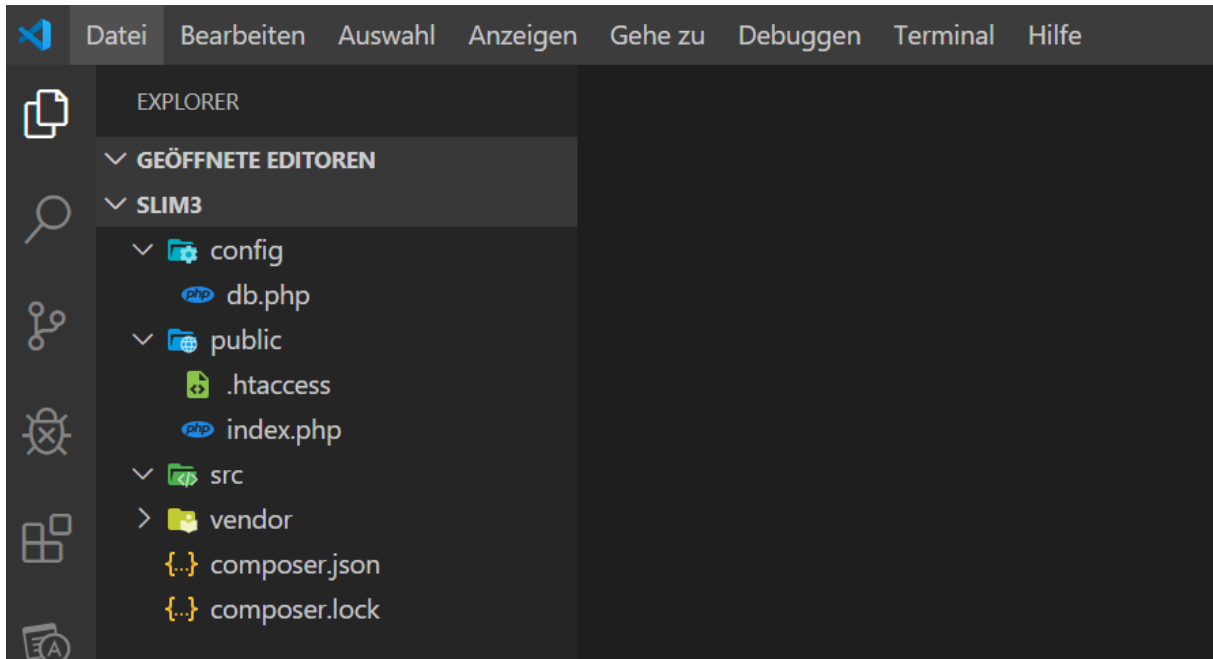
First Application

How to Install Slim

We recommend you install Slim with **Composer**. Navigate into your project's root directory and execute the bash command shown below. This command downloads the Slim Framework and its third-party dependencies into your project's `vendor/` directory.

```
composer require slim/slim:3.*
```

2) Lege nach der Installation eine Struktur an



3) Die Datenbank hier heißt:

Die Datenbank die verwendet wird ist „login“. Passwort: MD5 verschlüsselt

2 Datensätze: franz@gmail.com PSW: geheim

josef@gmail.com PSW: 1234

+ Optionen						
		login_id	email	psw		
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	1	franz@gmail.com	e8636ea013e682faf61f56ce1cb1ab5c
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	2	josef@gmail.com	1234

4) Daher die Datenbank-Verbindung in „db.php“

```
db.php x
config > db.php
1 <?php
2 /**
3  * Connect MySQL with PDO class
4  */
5 class db {
6
7     private $dbhost = 'localhost';
8     private $dbuser = 'root';
9     private $dbpass = '';
10    private $dbname = 'login';
11
12    public function connect() {
```

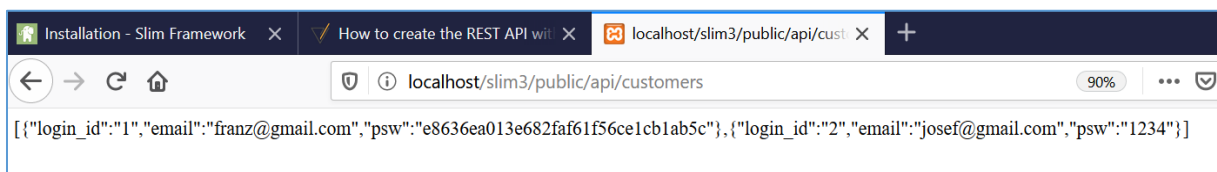
A)GET-Abfrage

A1)neue GET anlegen:

Erfolgt in der index.php unterhalb der vorhandenen ersten GET-Abfrage (die bereits vorhanden ist)

```
17
18 // create GET HTTP request
19 $app->get(['/api/customers', function( Request $request, Response $response){
20     // echo 'Unsere Logins';
21
22     $sql = "SELECT * FROM login";
23
```

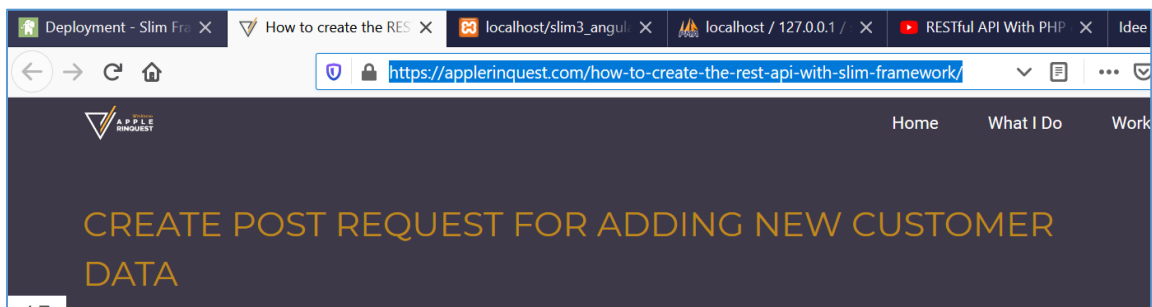
Ergebnis:



B)POST

B1)POST – neuen Datensatz anlegen

Weiter mit: <https://applerinquest.com/how-to-create-the-rest-api-with-slim-framework/>



Dafür benötigt man auch ein entsprechendes Formular. Dazu aber später.

Nun muss man den Code unterhalb der letzten Zeile der GET-Abfrage in die „index.php“ kopieren.

```
46     });
47
48 /**
49  * Add new customer data
50  */
51 // create POST HTTP request
52 $app->post('/api/post/add', function( Request $request, Response $response){
53
```

Ändere den Code für die Anpassung auf unsere Datenbank-Datensätze „login“:

+ Optionen				login_id	email	psw
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	1	franz@gmail.com	e8636ea013e
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	2	josef@gmail.com	1234

- email
- psw

Änderungen hier:

```

45 // get the parameter from the form submit
46 $name = $request->getParam('name');
47 $address = $request->getParam('address');
48
49
50 $sql = "INSERT INTO customers (name,address) VALUES(:name,:address)";
51

```

und

```

64 $stmt->bindParam(':name', $name);
65 $stmt->bindParam(':address', $address);
66

```

Ändern auf:

```

45 // get the parameter from the form submit
46 $email = $request->getParam('email');
47 $psw = $request->getParam('psw');
48
49
50 $sql = "INSERT INTO login (email,psw) VALUES(:email,:psw)";
51

```

und

```

64 $stmt->bindParam(':email', $email);
65 $stmt->bindParam(':psw', $psw);
66

```

B2) Testen mit <RESTED> bzw. Postman

Da wir nun kein Formular verwenden und schnell aber testen möchten, verwende z.B. Postman

- Methode: post
- Parameter befüllen

POST http://localhost/slim3/public/api/customers/add?email=test@gmx.at&psw=testen

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
email	test@gmx.at	
psw	testen	

Body Cookies Headers (7) Test Results Status: 200 OK Time: 35ms Size: 299 B Save R

```

1 {
2   "notice": {
3     "msg": "New Customer Added.."
4   }
}

```

Ergebnis betrachten:

- Entweder mit der Get-Abfrage von vorhin oder
- Direkt in der Datenbank nachsehen, ob ein Datensatz dazugekommen ist

	login_id	email	psw
<input type="checkbox"/>	1	franz@gmail.com	e8636ea013e682faf61f56ce1cb1ab5c
<input type="checkbox"/>	2	josef@gmail.com	1234
<input type="checkbox"/>	3	test@gmx.at	testen

Info in CMD:

composer info -i

```

C:\xampp\htdocs\slim3>composer info -i
You are using the deprecated option "installed". Only installed packages are shown by default now. The --all option can
be used to show all packages.
nikic/fast-route v1.3.0 Fast request router for PHP
pimple/pimple v3.2.3 Pimple, a simple Dependency Injection Container
psr/container 1.0.0 Common Container Interface (PHP FIG PSR-11)
psr/http-message 1.0.1 Common interface for HTTP messages
slim/slim 3.12.3 Slim is a PHP micro framework that helps you quickly write simple yet powerful web applica...

```

composer show --installed

```

C:\xampp\htdocs\slim3>composer show --installed
You are using the deprecated option "installed". Only installed packages are shown by default now. The --all option can
be used to show all packages.
nikic/fast-route v1.3.0 Fast request router for PHP
pimple/pimple v3.2.3 Pimple, a simple Dependency Injection Container
psr/container 1.0.0 Common Container Interface (PHP FIG PSR-11)
psr/http-message 1.0.1 Common interface for HTTP messages
slim/slim 3.12.3 Slim is a PHP micro framework that helps you quickly write simple yet po
C:\xampp\htdocs\slim3>

```