

3)Webshop in PHP – Code auslagern

Um die „index.php“ zu verkleinern und den Code den aktuellen modernen Schreibweisen anzupassen, werden bestimmte Code-Elemente in eigenen Dateien ausgelagert.

Dabei werden neue Ordner und Dateien angelegt und Code verschoben. Hauptsächlich geht es um Dateien für:

- Funktionen
- Templates
- Routes

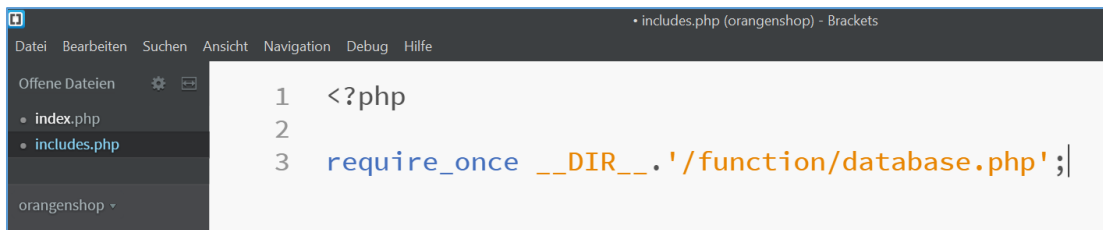
1)eigene „includes.php“

In diese Datei werden alle Einschließungen „require“ bzw. „include“ gesammelt. Es werden im Laufe des Webshops noch mehr als die eine vorhandene.

Erstelle im „root-Verzeichnis“ die Datei „includes.php“ und starte sie mit dem PHP-Tag:

```
<?php
```

Schneide aus der „index.php“ die Zeile 7 aus und füge sie hier ein:

A screenshot of a code editor window titled "includes.php (orangenshop) - Brackets". The editor shows three lines of PHP code: line 1: <?php, line 2: (empty), line 3: require_once __DIR__.' /function/database.php';. The left sidebar shows a file explorer with "index.php" and "includes.php" listed under "orangenshop".

```
1 <?php
2
3 require_once __DIR__.' /function/database.php';
```

Dafür lege in index.php die neue Verknüpfung ab.

A screenshot of a code editor showing two lines of PHP code: line 6: define('CONFIG_DIR',__DIR__.' /config');, line 7: require __DIR__.' /includes.php';.

```
6 define('CONFIG_DIR',__DIR__.' /config');
7 require __DIR__.' /includes.php';
```

```
require __DIR__.' /includes.php';
```

2)users.php

Aufgabe: Auslagern der Session, die die „userId“ liefert.

Dafür soll eine Datei im Ordner „function“ erstellt werden. Darin eine neue Funktion:

```
function getCurrentUserId(){}
```

```
Offene Dateien  ⚙️ 🗑️
├─ index.php
├─ includes.php
├─ user.php
└─ orangenshop ▾

1 <?php
2
3 ▼ function getCurrentUserId(){
4     |
5 }
```

Schneide aus der „index.php“ folgenden Zeilen aus, füge sie hier ein, damit diese neue Funktion das übernehmen kann. Die Zeile 10 mit der \$korbZahl muss bleiben.

```
8
9 $userId = 0;
10 $korbZahl = 0;
11
12 ▼ if(isset($_SESSION['userId'])){
13     $userId = (int) $_SESSION['userId'];
14 }
15
```

Vor dem Ende der Funktion muss man noch die „userId“ zurückliefern mit „return“:

```
1 <?php
2
3 ▼ function getCurrentUserId(){
4     $userId = 0;
5 ▼     if(isset($_SESSION['userId'])){
6         $userId = (int) $_SESSION['userId'];
7     }
8     return $userId;
9 }
```

return \$userId;

Noch 2 Schritte:

- In der „index.php“ muss man nun diese neue Funktion aufrufen: Zeile 15

```
10 $korbZahl = 0;
11
12 $userId = getCurrentUserId();|
13
14 $sql = "SELECT id, titel, beschreibung
```

- Die Verknüpfung mit „require“ muss noch in der neuen „includes.php“ eingetragen werden, da es nun die neue „user.php“ gibt.

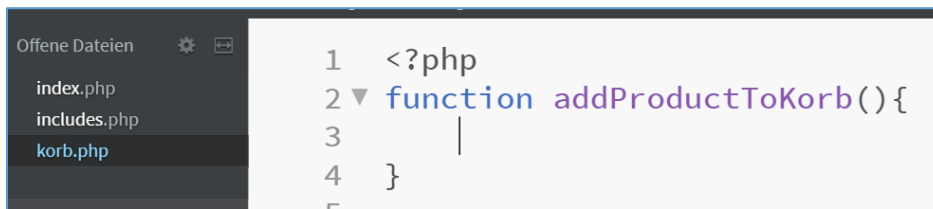
Kopiere die Zeile 3 und füge sie darunter ein. Ändere „database“ auf „user“.

```
Offene Dateien  ⚙️ 🗑️
├─ index.php
├─ includes.php
├─ user.php
└─ orangenshop ▾

1 <?php
2
3 require_once __DIR__.'function/database.php';
4 require_once __DIR__.'function/user.php';|
```

3)Warenkorb „korb.php“

Erstelle eine neue Datei im Ordner „function“ mit dem Namen „korb.php“ und erstelle eine Funktion darin:



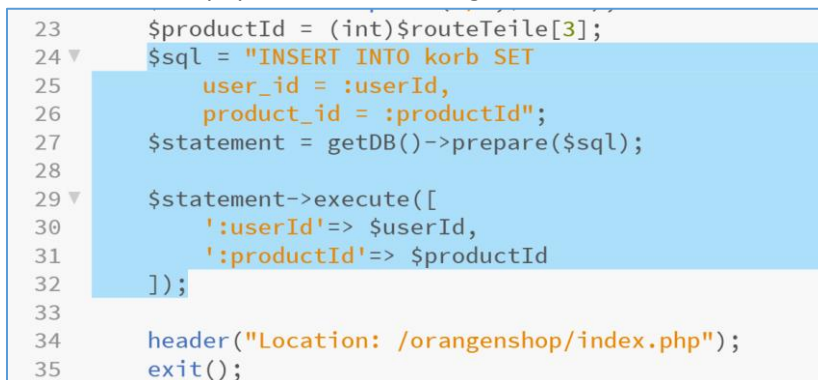
```
Offene Dateien
index.php
includes.php
korb.php

1 <?php
2 function addProductToKorb(){
3     |
4 }
```

```
function addProductToKorb(){}
```

In diese Funktion kommt nun ein Code-Teil aus der „index.php“, nämlich der, welcher für die Aufnahme von Produkten in die Datenbank erstellt wurde.

Öffne die „index.php“ und schneide folgende Zeile aus, um sie in die neue Datei hineinzuschieben:

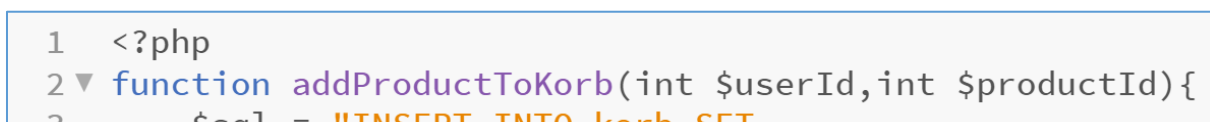


```
23 $productId = (int)$routeTeile[3];
24 $sql = "INSERT INTO korb SET
25     user_id = :userId,
26     product_id = :productId";
27 $statement = getDB()->prepare($sql);
28
29 $statement->execute([
30     ':userId'=> $userId,
31     ':productId'=> $productId
32 ]);
33
34 header("Location: /orangenshop/index.php");
35 exit();
```

Die Umleitung per „header“ bleibt in der index.php.

3a)Variablen

Da die Variablen \$userId und \$productId hier nicht zur Verfügung stehen von Beginn an, muss man diese als Parameter in den runden Klammern der Funktion eintragen, da sie Zahlen sind, muss man hier „int“ verwenden:



```
1 <?php
2 function addProductToKorb(int $userId,int $productId){
3     $sql = "INSERT INTO korb SET
```

3b)require erstellen

korb.php in der „include.php“ einfügen, damit man in der „index.php“ darauf zugreifen kann.



```
1 <?php
2
3 require_once __DIR__.' /function/database.php';
4 require_once __DIR__.' /function/user.php';
5 require_once __DIR__.' /function/korb.php';
```

3c) Aufrufen der Funktion

die neue Funktion in der „index.php“ aufrufen – Zeile 25 – dort, wo vorher der Code gestanden hat

```
24
25     addProductToKorb($userId, $productId);
26     header("Location: /orangenshop/index.php");
```

```
addProductToKorb($userId,$productId);
```

3d) zweite Funktion erstellen, für das Zählen

Erstelle in der „korb.php“ unterhalb eine 2. Funktion. Diese soll den Code aus der „index.php“ aufnehmen, der für das Zählen der Produkte im Warenkorb zuständig ist.

```
12 }
13
14 ▼ function countProductInKorb(){
15     |
16 }
```

```
function countProductInKorb(){}
```

Schneide aus der „index.php“ folgenden Code und füge ihn hier ein:

```
14 ▼ function countProductInKorb(){
15     $sql = "SELECT COUNT(id) FROM korb WHERE user_id =" . $userId;
16     $korbResult = getDb()->query($sql);
17
18     $korbZahl = $korbResult->fetchColumn();
19 }
```

Zwei wichtige Sachen sind nun sofort zu erledigen:

- Die Variable in den geschwungenen Klammern der neuen Funktion einsetzen
- Die Variable \$korbZahl muss mit einem „return“ eingefügt werden

```
14 ▼ function countProductInKorb(int $userId){
15     $sql = "SELECT COUNT(id) FROM korb WHERE user_id =" . $userId;
16     $korbResult = getDb()->query($sql);
17
18     $korbZahl = $korbResult->fetchColumn();
19     return $korbZahl;
20 }
```

- In der „index.php“ muss nun auch die neue Funktion angeführt werden. Am besten dort, wo man gerade die Zeilen ausgeschnitten hat.
\$korbZahl = countProductInKorb(\$userId);

```

28 }
29
30 $korbZahl = countProductInKorb($userId);
31
32 require __DIR__.'/templates/main.php';

```

- Dafür benötigt man die Variable, in der damals ein Wert initialisiert wurde, nämlich \$korbZahl nicht mehr und man kann diese löschen.

```

6 define('CONFIG_DIR',__DIR__.'/config');
7 require __DIR__.'/includes.php';
8
9 $korbZahl = 0;

```

4)product.php

Auslagern des Codes bezüglich „product“.

Erstelle im Ordner „functions“ eine Datei „product.php“. Erstelle dort eine neue Funktion „getAllProducts“



```

Offene Dateien
index.php
includes.php
product.php
orangenshop ▾

1 <?php
2
3 ▼ function getAllProducts(){
4
5 }

```

function getAllProducts(){}

Schneide aus der „index.php“ folgende Zeile aus und übertrage es hier her:
Darunter erstelle den return vom „\$result;“.

```

1 <?php
2
3 ▼ function getAllProducts(){
4     $sql = "SELECT id,titel,beschreibung,preis,anzahl,produktbild FROM products";
5
6     $result = getDB()->query($sql);
7
8     return $result;
9 }
10

```

Noch 2 Schritte:

- In der „index.php“ muss man nun diese neue Funktion aufrufen und der Variablen \$result zuordnen:

```
$result = getAllProducts();
```

```
11 $userId = getCurrentUserId();  
12 $result = getAllProducts();  
13
```

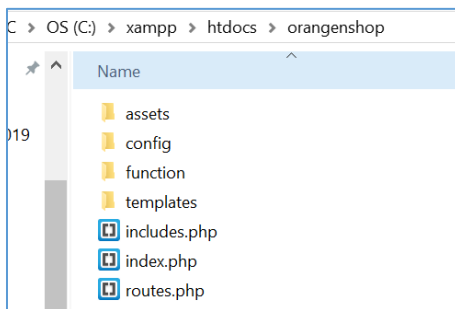
- Die Verknüpfung mit „require“ muss noch in der neuen „includes.php“ eingetragen werden.

```
3 require_once __DIR__.' /function/database.php';  
4 require_once __DIR__.' /function/user.php';  
5 require_once __DIR__.' /function/korb.php';  
6 require_once __DIR__.' /function/product.php';
```

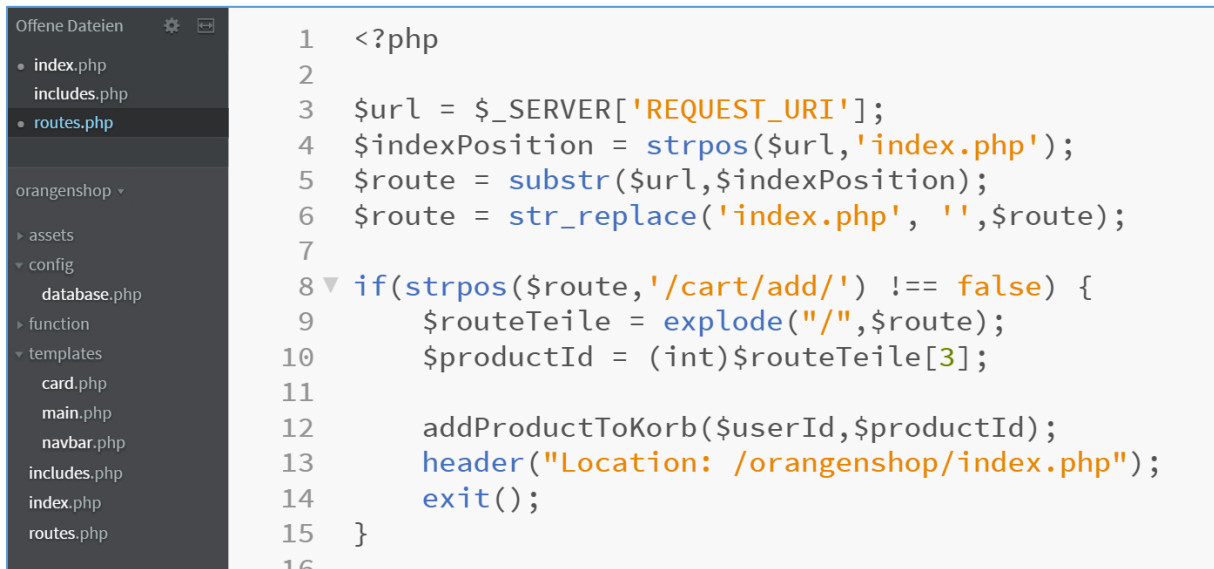
5) routes.php

Hier sollen sich später alle definierten URLs befinden.

Erstelle eine neue „routes.php“ im Grundverzeichnis, neben index und includes.



Diese soll ebenfalls Code aus der „index.php“ erhalten:



Füge die passende „require“ in die „includes.php“ ein: Beachte jedoch, dass hier nicht der Ordner „function“ nötig ist.

```

6 require_once __DIR__.'/function/product.php';
7 require_once __DIR__.'/routes.php';

```

Nicht vergessen: auch die Variable „\$userId“ muss man hier bekannt machen und daher aus der „index.php“ folgende Zeile kopieren und hier einfügen:

```

9 $userId = getUserId();

```

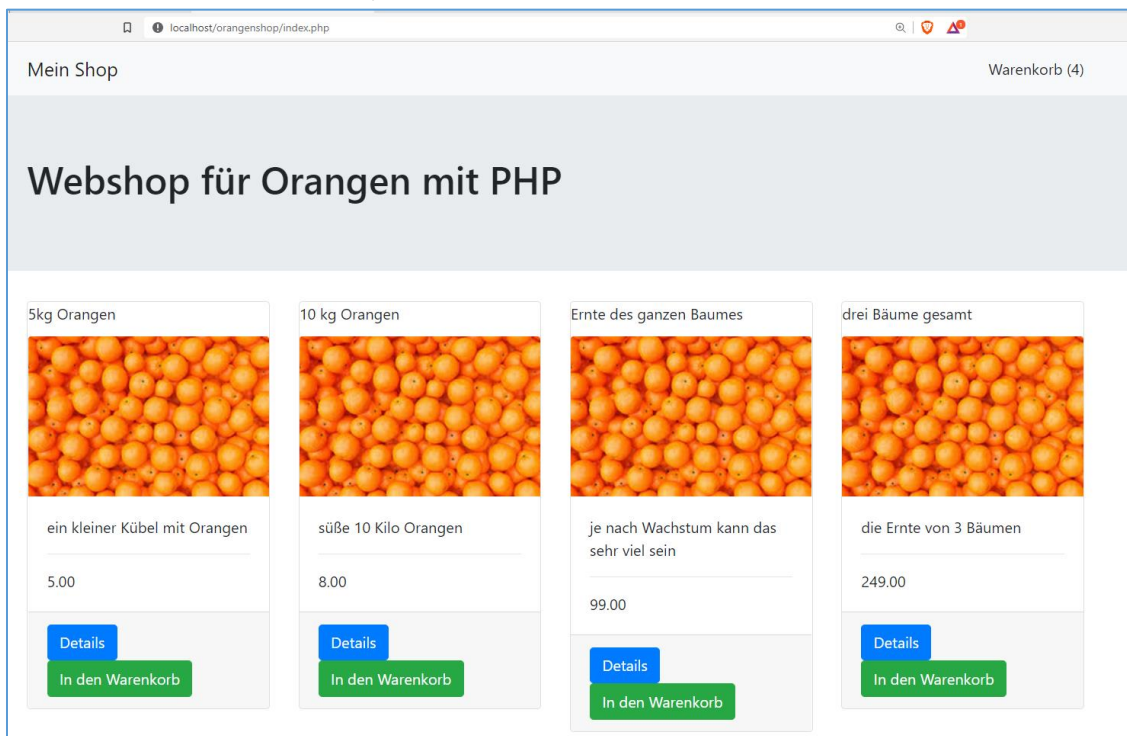
Ergebnis:

```

6 $route = str_replace('index.php', '', $route);
7
8 $userId = getUserId();
9
10 if(strpos($route, '/cart/add/') !== false)

```

Zum Abschluss nochmal testen, ob die Seite weiterhin aufrufbar ist:



In der Index ist nur mehr folgendes vorhanden:

```
1 <?php
2 session_start();
3 error_reporting(-1);
4 ini_set('display_errors','0n');
5
6 define('CONFIG_DIR',__DIR__.'/config');
7 require __DIR__.'/includes.php';
8
9 $userId = getCurrentUserId();
10 $result = getAllProducts();
11
12 $korbZahl = countProductInKorb($userId);
13
14 require __DIR__.'/templates/main.php';
15 ?>
```

Quelle: <https://www.youtube.com/watch?v=XGNeqpW8k5A>