

# Datenbanken & Webtechnologien

Marcel Remmy M. ENG.

## **PRAKTIKUM MEILENSTEIN 2**

### **PAKET 2**

SUPPORT: KW45

ABNAHME: KW47

### **PAKET 3**

SUPPORT: KW46

ABNAHME: KW47

Da es ein paar Fragen zur Abgabemodalität gabe:  
Die Abgabe nehmen Sie ab Donnerstag, den 14.11. **bis spätestens  
Ende Samstag, den 16.11.2019** vor.

- Sowohl die SQL-Datei aus Paket 2 als auch die ZIP-Datei mit der Lösung aus dem vorliegenden Paket 3 sind Gegenstand der Abgabe.
- Die SQL-Datei muss ohne DROP DATABASE auskommen und 2x ausführbar sein.
- Zweierteams werden wieder angegeben und jedes Team-Mitglied muss gesondert den Upload der Abgabe vornehmen.

Das Dossier hat denselben Stellenwert wie Ihre Lösungen in SQL und PHP, also pflegen Sie es entsprechend.

# Meilenstein 2 | Paket 3

In diesem Paket werden Sie die vorbereiteten HTML-Templates aus Meilenstein 1 und die sinnvoll vorbereiteten Fragmente mittels Layout-Struktur und Referenzen von Redundanzen befreien.

Sie lernen, wie Sie GET- und POST-Parameter entgegennehmen und Daten aus Ihrer Datenbank auf den Webseiten anzeigen können.

Machen Sie sich dazu zunächst mit der verwendeten Programmiersprache PHP vertraut. Zum Senden von SQL-Abfragen an die MariaDB-Server per PHP nutzen wir einen entsprechenden Treiber: `mysqli`.

## Vorbereitung : Projekt anlegen mit composer init

Ab diesem Paket werden Sie mit PHP arbeiten und wie in jeder Programmiersprache kommt es zu der Situation, dass Sie auf bestehenden, fremden Code zurückgreifen möchten. Das wird in späteren Aufgaben nachvollziehbar. Damit Sie auf ein bequemes Paketmanagement zurückgreifen können, werden wir im Praktikum [composer](#) einsetzen.

### Mit der Powershell zum Ordner `xampp/htdocs/` wechseln und darin einen neuen namens "M2" anlegen

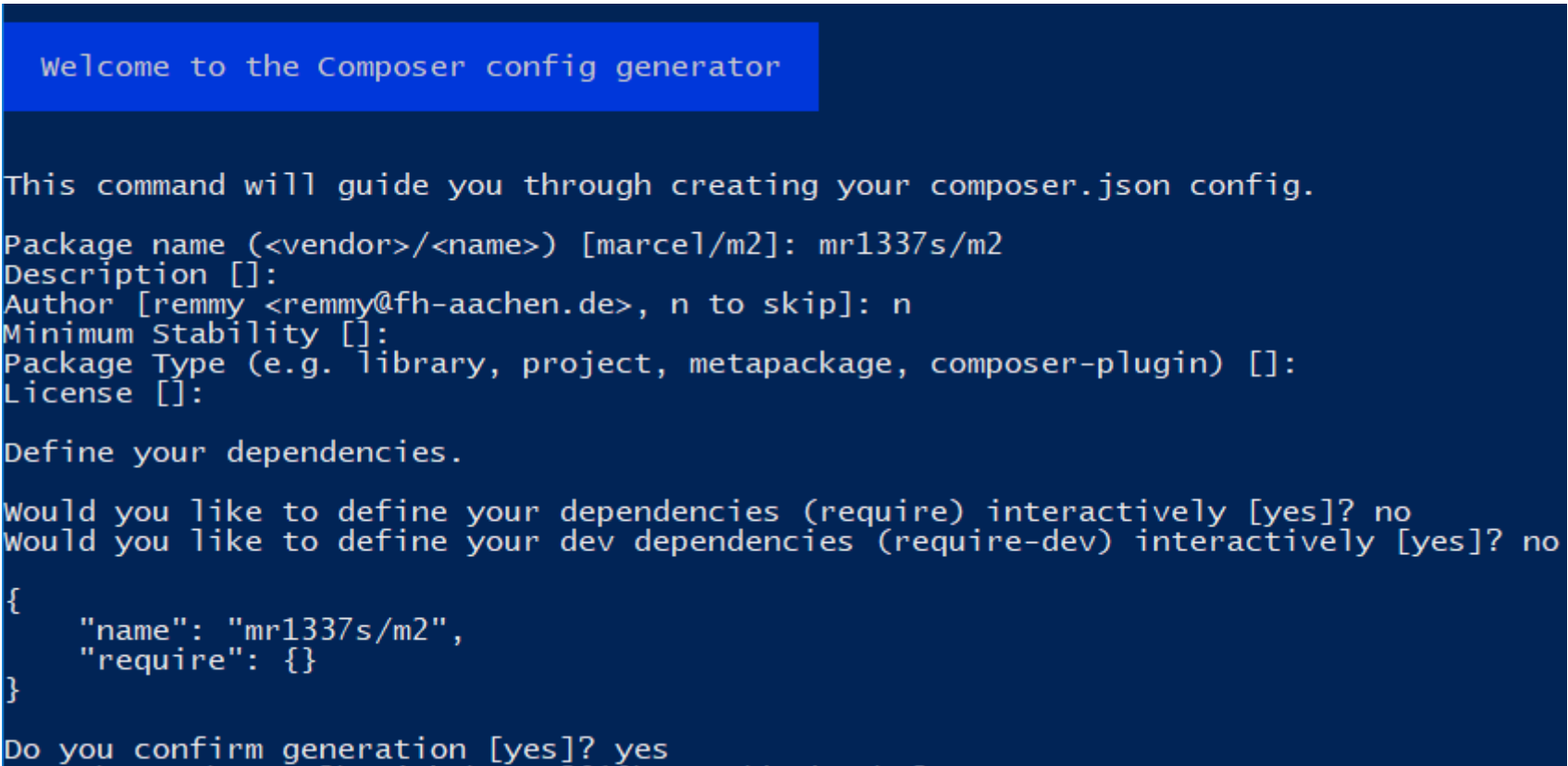
1. Starten Sie die Powershell (Kommandozeile in Windows) mit `WIN+R` und geben Sie im Ausführen-Dialog "powershell" ein. Daraufhin startet sich die Powershell.
2. Navigieren Sie zum Ordner mit dem Befehl `cd zielpfad` (change directory).
3. Mit `mkdir M2` wird der neue Ordner erzeugt. Wechseln sie mit `cd M2` anschließend in den Ordner.

### Mit composer ein neues PHP Projekt anlegen

1. Wenn Sie in der Vorbereitung das ZIP File aus Ilias heruntergeladen haben, liegt Composer als PHP Archive im Ordner `xampp/composer/` vor. Wenn nicht, laden Sie Composer<sup>1</sup> selbst herunter (oder nutzen Sie die Anleitung "Composer-Projekte mit PhpStorm" aus Ilias). Nutzen Sie die folgende Zeile in der Powershell, um ein neues PHP Projekt mit composer anzulegen:

```
php ../../composer/composer.phar init
```

2. Geben Sie für *Package name (/)* Ihr Iliaskürzel als vendor und "m2" als Namen an. `<Ihr Ilias-Kürzel> / m2`
3. Die folgenden Fragen können Sie unbeantwortet lassen, bzw mit "no" beantworten, bis auf die letzte: *Do you confirm generation? yes*
  - Daraufhin wird composer die benötigte JSON-Datei anlegen.



Info:

Sie können Composer auch installieren bzw systemweit verfügbar machen. Dann können Sie statt des PHP Archivs einfach `compose` nutzen. Aus dem Aufruf `php ../../composer/composer.phar init` wird dann `php compose init`, das geht jedoch nicht portabel (und nicht am Terminal PC). Deshalb werden die Befehle immer die `composer.phar` Version zeigen.

### Erste Abhängigkeit verwalten: DotEnv

Mit composer können Sie nun PHP Code aus Repositories wie Packagist herunterladen. Da Sie im Verlauf des Pakets für die Verbindungsinformationen zur Datenbank eine Konfigurationsdatei anlegen sollen, wird PhpDotEnv hier vorgestellt.

Mit dem Befehl `php ../../composer/composer.phar vlucas/phpdotenv` ergibt sich eine Ausgabe wie folgt:

```
php ../../composer/composer.phar require vlucas/phpdotenv
Using version ^3.5 for vlucas/phpdotenv
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 3 installs, 0 updates, 0 removals
  - Installing symfony/polyfill-ctype (v1.12.0): Loading from cache
  - Installing phpoption/phpooption (1.5.0): Loading from cache
  - Installing vlucas/phpdotenv (v3.5.0): Loading from cache
Writing lock file
Generating autoload files
```

Dadurch wurde im Ordner `M2/vendor/` PHP Code heruntergeladen, der es ermöglicht, bequem auf `.env` Konfigurationsdateien zuzugreifen, siehe [Aufgabe 3.2](#).

## Aufgabe 3.1 : Redundanzen verringern

Sie erkennen, dass auf vielen Seiten das selbe Markup steht. Verringern Sie diese Redundanz mittels der PHP-Anweisung `include`. Bedenken Sie, dass Sie dazu Ihre aufgerufene Resource in eine PHP-Datei umwandeln müssen (Umbenennen).

- Inkludieren Sie die Navigationselemente oben und unten, indem Sie die Menüs in einzelne Dateien (z.B. *NavOben.php* und *NavUnten.php*)auslagern. Es empfiehlt sich ein neuer Ordner, um die Übersicht zu behalten.
- Wenn dieser Ordner *snippets* heißt, können Sie den Inhalt per `<?php include('snippets/NavOben.php'); ?>` wieder einfügen.<sup>2</sup>
- Überlegen Sie, welche Fragmente aus Ihren HTML-Seiten Sie noch auslagern möchten. ✍ Beschreiben Sie kurz im Dossier, welche Elemente Sie für sinnvoll hielten.
- Damit Sie auf allen Seiten dieselbe Datei für das Menü oben inkludieren können, nutzen Sie PHP um festzustellen, welcher Menüpunkt nun aktiv und damit nicht verlinkt ist. Sie hatten das in M1 vorbereitet, nun kommt die "Template Logik" mit ins Spiel. In der Globalvariable `$_SERVER['REQUEST_URI']` finden Sie, welche Seite aufgerufen wurde. Beispiel: Sie fügen auf allen Seiten die Linkliste aus dem Kopf ein. Wenn die aktuelle Seite "Produkte.php" ist, soll der Link "Mahlzeiten" nicht nutzbar bzw. aktiv sein.

Bearbeiten Sie erst Aufgabe 3.1 bevor Sie fortfahren!

## Aufgabe 3.2 : Datenbankzugriff für die Zutatenliste

Sie wissen nun, wie Sie HTML und PHP vermischen können. Immer, wenn Sie programmatisch Daten erzeugen oder anfordern, öffnen Sie das PHP Tag, und nach Ihrem Code schließen Sie es wieder. Realisieren Sie nun den ersten Datenbankzugriff auf Ihre neue Datenbank.

Eine Verbindung zur Datenbank können Sie mittels `mysqli_connect()` erreichen<sup>3</sup>. Sie müssen sowohl die Adresse des DB-Servers, als auch den Port angeben. Darüberhinaus müssen Sie den Datenbankbenutzer und sein Passwort angeben, um Zugriff auf die Resource zu erhalten.

Nutzen Sie eine der in der Vorlesung vorgestellten Techniken, die SQL-Abfragen per PHP an die Datenbank zu stellen. Nutzen Sie (in diesem Meilenstein) **kein ORM** wie Propel, Doctrine oder Eloquent.

Es empfiehlt sich, in HeidiSQL immer die SQL Query zu entwickeln und die resultierenden Daten zu betrachten, bevor Sie in PHP die Daten dann entgegennehmen.

Prüfen Sie in jedem Fall, dass die Verbindung zur Datenbank funktioniert hat (gab es einen `mysqli_connect_error` ?).

```
$query = 'SELECT ID,Name FROM TableY;'; // Ihre SQL Query aus HeidiSQL

$remoteConnection = mysqli_connect('host', 'user', 'password', 'databasename');

if (mysqli_connect_errno()) {
    printf("Konnte nicht zur entfernten Datenbank verbinden: %s\n", mysqli_connect_error());
    exit();
}

if ($result = mysqli_query($remoteConnection, $query)) {
    while ($row = mysqli_fetch_assoc($result)) {
        // $row['ID'] und $row['Name'] stehen aus der Query zur Verfügung
        echo '<li id="id-'. $row['ID'] .'">'. $row['Name'] . '</li>';












    }
}

mysqli_close($remoteConnection); // daran denken, die Verbindung wieder zu schließen wenn sie nicht mehr benötigt ist.
```

## Zutatenliste

Generieren Sie eine Seite, die im von Ihnen geschaffenen Layout, die Zutaten allesamt auflistet. Diese Seite soll `Zutaten.php` heißen und im Fußbereich der bisher bestehenden Seiten verlinkt werden. Wenn Sie Aufgabe 3.1 bereits umgesetzt haben, ist das eine kleine Änderung an der Datei `snippets/NavUnten.php`.

- Die Liste kann als `<ul>` oder, korrekter als `<table>` umgesetzt werden.
- Zeigen Sie zu jeder Zutat den Namen und die Eigenschaften an (also ob Bio, Vegan, Vegetarisch und/oder Glutenfrei)
  - Erinnern Sie sich, dass Sie hierzu mehrere Möglichkeiten haben (CSS-Klasse mit `background`, `<img />`, oder ein SVG-Icon)
- Wenn die Liste steht, ändern Sie die Query so ab, dass zuerst die Bio-Zutaten alphabetisch, und dann die Nicht-Bio-Zutaten alphabetisch ausgegeben werden.
- Geben Sie in der Überschrift `Zutatenliste` in Klammern die Anzahl der gefundenen Zutate an.
  - 🔗 Welche Möglichkeiten für diese Umsetzung kennen Sie?
- Generieren Sie für den Benutzer auch einen Link, mit dem er eine Zutat auch bei der Suchmaschine Ihrer Wahl suchen kann (erinnern Sie sich Ihr Suchformular aus Meilenstein 1: Sie haben mit der GET-Methode Links generiert)
- Bleibt man mit dem Mauszeiger über dem Link sollte gezeigt werden, dass man so nach der Zutat im Web suchen kann (siehe Bild).

Zutat	Vegan?	Vegetarisch?	Glutenfrei?
 <a href="#">Amaranth</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
 <a href="#">Champignons</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
 <a href="#">Fenchel</a> <div>Suchen Sie nach Fenchel im Web</div>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
 <a href="#">Hanfmehl</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
 <a href="#">Heilbutt</a>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
 <a href="#">Kurkumin</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
 <a href="#">Lachs</a>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
 <a href="#">Paprika</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
 <a href="#">Sojasprossen</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
 <a href="#">Weizenmehl</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 <a href="#">Zitronensäure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Aal</a>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Alginat</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Barsch</a>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Branntweinessig</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

- Tragen Sie zum Testen ein paar Zutaten ein.
- Ab dem 11.11.19 finden in der Datenbank "public" eine Tabelle **Zutaten**, deren Daten Sie in Ihre eigene Datenbank aus Paket 2 kopieren sollen. 🔗 Notieren Sie die dazu erforderliche SQL-Queries im Dossier.

## Aufgabe 3.3 : Dynamische Produkt- und Detailseiten

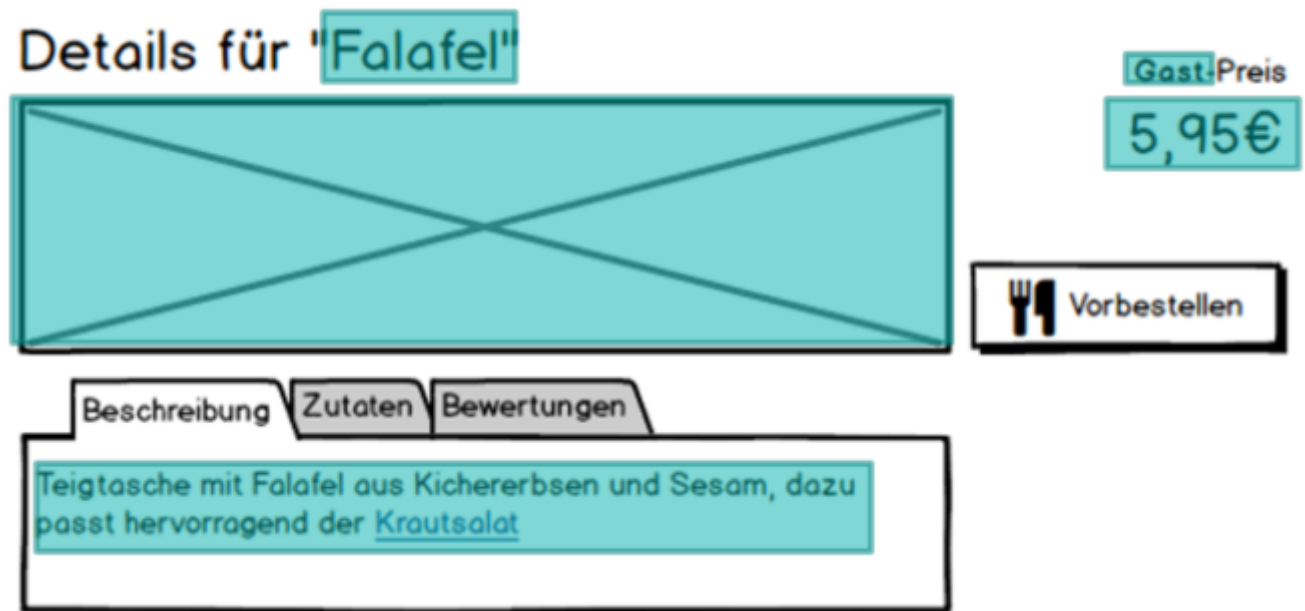
In PHP-Skripten können Sie per HTTP Request übertragene Daten wie in der Vorlesung gezeigt entgegennehmen. Nutzen Sie das, z.B. um die Mahlzeiten-ID an die `Detail.php` per GET zu übertragen. Sie legen den Feldnamen selbst fest. Prüfen Sie, Sie beim Aufruf von `http://localhost:8080/M2/Detail.php?id=4` im PHP-Skript per `$_GET['id']` auch die 4 erhalten.

Falls noch nicht geschehen, ändern Sie nun die Endungen der Dateien von `.html` auf `.php` und passen die Links in den Linklisten entsprechend an.

## Detail.php

Bereits in Meilenstein 1 haben Sie im Dossier festgehalten, dass nur einige Teile in den einzelnen Detailseiten für die einzelnen Mahlzeit-Informationen zu ändern wären. All diese HTML-Fragmente werden Sie nun per PHP manipulieren.

Hierzu brauchen Sie eine geeignete SQL-Query, die Ihnen die benötigten Daten für die Mahlzeit mit der angegebenen ID bereitstellt. Entwickeln Sie diese in HeidiSQL und notieren Sie sie.



- Laden Sie alle erforderlichen Daten mit `mysqli_query()` und `fetch_assoc()`
- Wird eine ungültige oder keine ID an die Seite übermitteln, leiten Sie den Benutzer zurück zur Seite `Produkte.php`. Sie haben in Meilenstein 1 eine Möglichkeit zur Weiterleitung kennengelernt. Finden Sie einen Weg, das bereits mit PHP zu tun (erinnern Sie sich daran, wie die HTTP Header aufgebaut sind und was ein Redirect auslöst).
- Nicht angemeldete Benutzer sehen immer den Gastpreis. Aktuell existiert noch keine Loginmöglichkeit, deshalb zeigen Sie immer den Gastpreis. Auch dieser sollte aus der Datenbank kommen.
- Optional: Mahlzeiten haben Zutaten. Legen Sie auch die Zutatenliste in der Detailseite dynamisch an. In den meisten Fällen wird dies eine weitere SQL-Query mit sich bringen.

## Produkte.php

Die Mahlzeiten-Seite ändert sich nun: Entfernen Sie die statischen acht Mahlzeiten aus Ihrem Prototyp. Stattdessen wird nun für jede gefundene Mahlzeit in Ihrer Datenbank das Markup dynamisch erzeugt.

`Produkte.php` soll folgende beiden Felder entgegennehmen können:

- `limit` : Falls Sie ein Limit angeben, zeigen Sie maximal so viele Mahlzeiten, wie in Limit angegeben.
- `avail` : Falls dieses Feld einen Wert besitzt ( `true` ), soll die Seite nur verfügbare Mahlzeiten anzeigen (das abgeleitete Attribut `Mahlzeiten.Verfügbar` ist nicht `false` )<sup>4</sup>.

### Beispiel

- Haben Sie fünf Mahlzeiten testweise in die Tabelle eingetragen, sollte Ihre Seite `Produkte.php` auch fünf Mahlzeiten anzeigen.
- `Produkte.php?avail=1&limit=4` zeigt von allen verfügbaren Speisen vier Stück an

## Bilder

Sie haben für Meilenstein bereits eines oder mehrere Bilder für jede Mahlzeit verwendet. Diese Bilder können Sie gerne wiederverwenden. Wenn noch Bilder zu suchen oder zu erzeugen sind: spätestens jetzt ;)

Die Tabelle Mahlzeiten hat Bilder in Binärform in der Datenbank<sup>5</sup>. Falls Sie noch keine Binärdaten vorliegen haben, könnten Sie die Bild-ID nutzen um einen Dateinamen im Attribut `src` zu generieren. Das könnte generell Ihre Strategie sein für Entitäten in der Tabelle `Bilder`, die keine Binärdaten aufweisen.

Mit den PHP-Funktionen `base64_encode()` und `base64_decode()` können Sie die Binärdaten umwandeln.

### Beispiel

- Mahlzeit ID 4 ist verknüpft mit Bild ID 21
- Sie lesen die Attribute der Tabelle Bild aus und evaluieren, ob es Binärdaten gibt.
- Falls ja: Nutzen Sie die Base64 Repräsentation der Binärdaten für das Bild: ``

- Falls nein: Beim Generieren des HTML für die Mahlzeit nutzen Sie `` . Sobald alle Bilder in Binärformat vorliegen, entfernen Sie diese Rückfallstrategie.

## Konfiguration in externer Datei

Bereits in [Aufgabe 3.0](#) haben Sie PhpDotEnv heruntergeladen. Dadurch können Sie eine Datei namens `.env` mit Schlüssel-Wert-Paaren anlegen, die z.B. folgenden Inhalt haben könnte:

```
DB_HOST="149.201.88.100"
DB_PASS="mypasswordissafe"
DB_USER="villainremmy"
DB_PORT=3306
DB_NAME="thatnewdatabase"
```

In Ihren Programmen können Sie von nun an die Schlüssel abfragen und erhalten die zugehörigen Werte. Sobald Sie `$dotenv->load();` ausgeführt haben, können Sie per `getenv('DB_NAME')` etwa den Datenbanknamen auslesen. Das ist wesentlich besser, als die Angaben an vielen Stellen in Ihrem Code zu suchen.

```
$dotenv = Dotenv\Dotenv::create(__DIR__, '.env');
$dotenv->load();
$dotenv->required(['DB_HOST', 'DB_NAME', 'DB_USER', 'DB_PASS', 'DB_PORT']);

$remoteConnection = mysqli_connect(
    getenv('DB_HOST'),
    getenv('DB_USER'),
    getenv('DB_PASS'),
    getenv('DB_NAME'),
    (int) getenv('DB_PORT')
);
```

Denken Sie daran, am Beginn Ihrer PHP-Dateien die Direktive `require __DIR__ . '/vendor/autoload.php';` zu nutzen. Nur dann werden die Pakete, die Sie mit Composer hinzugefügt haben, auch geladen.

## Bedingung für die Abgabe: Die Ausgabe Ihrer PHP-Dateien muss valides HTML sein

Das kennen Sie aus dem ersten Meilenstein. Diesmal achten Sie bereits bei der Bearbeitung darauf, dass Ihr PHP valides HTML generiert.

Sie können das wie folgt prüfen:

- Lösen Sie einen GET Request mit Ihrem Browser aus, in dem Sie das PHP Skript anfordern. Das Skript generiert HTML und sendet es Ihrem Browser.
- Nutzen Sie **STRG+U** oder "Quelltext anzeigen", um das HTML zu sehen und kopieren Sie das Markup.
- Fügen Sie es in den Validator ein und beheben Sie eventuell gezeigte Fehler selbstständig.

## Troubleshooting / Typische Probleme

### (ERR\_CONNECTION\_REFUSED): localhost hat die Verbindung abgelehnt

Der Apache httpd Prozess läuft nicht, oder nicht unter dem angegebenen Port. Prüfen Sie dies im XAMPP Control Panel.

**(HY000/2002): Ein Verbindungsversuch ist fehlgeschlagen, da die Gegenstelle nach einer bestimmten Zeitspanne nicht richtig reagiert hat, oder die hergestellte Verbindung war fehlerhaft, da der verbundene Host nicht reagiert hat.**

Prüfen Sie, ob der Host/IP und der Port mit den im Praktikum erhaltenen Zugangsdaten übereinstimmen. Ihr mysql konnte sich nicht dem Server verbinden. Beachten Sie: Der Server ist nur per VPN von ausserhalb der FH und im Eduroam zu erreichen.

### (HY000/1045): Access denied for user 'username'@'ip' (using password: YES)

Die Fehlermeldung sagt es schon: Die Login-Daten stimmen nicht. Prüfen Sie erneut, ob DB-Nutzer und DB-Passwort stimmen. Ein Datenbankserver hat jedoch geantwortet.

### Too many connections

Achten Sie darauf, jede geöffneten Datenbankverbindung (unerheblich ob in PHP oder in SQL-Clients) auch wieder zu schließen. Die DB-Server sind so angelegt, dass Sie bis zu sechs gleichzeitige Verbindungen für Ihren Benutzer zulassen. Doch jede Verbindung kann mehrere Minuten offen bleiben, wenn sie nicht explizit geschlossen wird, und wenn alle gleichzeitigen Verbindungen aufgebraucht ist müssen Sie das abwarten.

Wie immer: Nutzen Sie auch das FAQ in Ilias, in dem auftretende Probleme gesammelt geklärt werden.



FH Aachen Fachbereich Elektrotechnik und Informationstechnik

Marcel Remmy  
Büro H215  
Eupener Straße 70  
52064 Aachen






TEL | +49. 241. 6009 52198

✉ | remmy@fh-aachen.de

🌐 | <http://fh-aachen.de/remmy>

🕒 | Sprechstunde nach Vereinbarung

## Referenzen

1. Composer für PHP herunterladen: <https://getcomposer.org/download/> .
2. Dort, wo diese Inklusion geschieht, wird der Inhalt der angegebenen Datei platziert. (Siehe Folie PHP Seite 49) .
3. mysqli\_connect() <https://www.php.net/manual/de/function.mysqli-connect.php> .
4. Wie wissen Sie, ob eine Mahlzeit verfügbar ist? Grundregel: Wenn der Vorrat größer 0 ist, ist die Mahlzeit verfügbar .
5. Base64 in HTML-Bildern: [https://wiki.selfhtml.org/wiki/Grafik/Grafiken\\_mit\\_Data-URI](https://wiki.selfhtml.org/wiki/Grafik/Grafiken_mit_Data-URI) .