

PHP8 - Fortgeschrittenenkurs

Dr.sc.nat. Michael J.M. Wagner, New Elements*

Revision 1.7



*michael@wagnertech.de

Inhaltsverzeichnis

1	Einführung	4
1.1	Was ist PHP?	4
1.2	Beispielprogramm Bücherei	4
2	Fehlerbehandlung	5
3	Modularisierung	6
4	Objektorientierung	7
4.1	Klassen	7
4.2	Vererbung	10
4.3	Abstrakte Klassen, Schnittstellen und Traits	11
5	Weitere Themen	12
5.1	Namensräume	12
5.2	Reguläre Ausdrücke	13
6	XML	14
6.1	SimpleXML	14
6.2	DOMDocument	15
7	Ajax	16
7.1	Hallo Ajax	16
7.2	Parameter senden	17
7.3	XML-Datei lesen	17
8	PHP 8	17
9	Quellen	18

IT-Schulungen.com Portfolio

IT-Schulungen.com ist eines der führenden, herstellerunabhängigen Seminarportale von Schulungen rund um die Informationstechnologie (IT) und das IT-Management. Seit über 15 Jahren ist IT-Schulungen.com eine anerkannte Anlaufstelle für viele Unternehmen und Behörden, wenn es um die Durchführung von DACH-weiten Schulungen geht.

- | | | |
|---|---|--|
| <ul style="list-style-type: none"> • Applikationsserver / Middleware • Business Intelligence • Business-Skills und Führung • Cloud • CRM • Datenbanken • eBusiness | <ul style="list-style-type: none"> • ERP-Systeme • IT Management • IT-Recht / Lizenzierung • ITIL • Mobile • Multimedia • Office | <ul style="list-style-type: none"> • Open Source • Portale • SAP® • Security • Serversysteme • Softwareentwicklung • Systemmanagement |
|---|---|--|

www.IT-Schulungen.com

New Elements GmbH | IT-Schulungen.com

Zertifizierungen & Partnerschaften



www.IT-Schulungen.com

New Elements GmbH | IT-Schulungen.com

1 Einführung

1.1 Was ist PHP?

PHP ist die Abkürzung für PHP Hypertext Preprocessor. PHP ermöglicht Entwicklern die Erzeugung dynamischer Internetseiten, mit denen sogenannte Web Applications erstellt werden, wie z. B. E-Commerce-Systeme, Chats oder Foren. Im Unterschied zu statischen Internetseiten kann sich der Inhalt aufgrund von Aktionen des Benutzers oder neuer Basisinformationen, die z. B. aus Datenbanken stammen, jederzeit ändern.

PHP unterstützt insbesondere die einfache Auswertung von Formularen, mit denen ein Benutzer Daten an eine Website senden kann. Es ermöglicht die Zusammenarbeit mit vielen verschiedenen Datenbanksystemen. Die weitaus meisten PHP-Entwickler setzen das Datenbanksystem MySQL ein.¹

PHP bietet im Vergleich zu anderen Programmiersprachen viele Vorteile. Als wichtigste Gründe für die Nutzung von PHP sind zu nennen:

- Es wurde zur Entwicklung von Internetanwendungen erschaffen.
- Es ermöglicht die einfache Entwicklung von Programmen.
- Es unterstützt verschiedene Plattformen.
- Es arbeitet sehr gut mit dem verbreiteten Apache Webserver zusammen, ...²

1.2 Beispielprogramm Bücherei

Als Beispielprogramm für diesen Kurs wird eine „Büchereiverwaltung“ implementiert. Um nicht bei Null beginnen zu müssen, steht bereits eine einfache Implementierung des Geschäftsvorfall „Neues Medium anlegen“ zur Verfügung. Diese Implementierung enthält folgende Dateien:

- `add_medium.html`: Formular zur Dateneingabe
- `add_medium.php`: verarbeitet den Request, fügt einen Datensatz der Datei. Im Erfolgsfall wird über einen *redirect* `show_media.php` aufgerufen. `/var/www/medien.csv` hinzu.
Anmerkung: Falls der Fehler „Datei nicht beschreibbar.“ auftritt, muss dieses Verzeichnis mit `sudo chown www-data:www-data /var/www` für diesen Nutzer schreibbar gemacht werden.
- `show_media.php`: zeigt alle in der Datei befindlichen Medien.
- `error_view.php`: wird im Fehlerfall aufgerufen.
- `upload.html`: wählt lokal eine Mediendatei aus und kopiert sie auf den Server.
- `upload.php`: verarbeitet den Request.

¹Theis: S. 18.

²Theis: S. 18.

Zur Beachtung: Das Beispiel implementiert das POST/REDIRECT/GET-Muster: Bei datenverändernden Anfragen macht der datenverarbeitende Request keinerlei HTML-Ausgaben. Nach der Bearbeitung leitet er zu einen GET-Request weiter. Damit wird erreicht, dass die Anwendung „navigierbar“ wird.

Aufgabe:

- Legen Sie in Eclipse ein Projekt *Bucherei* im *workspace* `workspace_php` an.
- Kopieren Sie die Dateien in das Projekt (`http://wagnertech.de/public/php/Bucherei.tgz`).
- Testen Sie den Code. Die Einstiegs-URLs sind:
`http://localhost/kurs/Bucherei/add_medium.html`, `upload.html`

2 Fehlerbehandlung

Zur Fehlerbehandlung gibt es verschiedene Ansätze:

- Rückgabewerte vom Typ `int`
- Rückgabewerte eines speziellen Fehlertyps
- Exceptions
- Eigene Routine zur Ermittlung des Fehlerstatus

Vorteil der Rückgabewerte:

- Einfach in der Behandlung, wenn an Ort und Stelle auf den Fehler reagiert werden kann.

Vorteil der Exceptions:

- Fehlerbehandlung stört nicht den logischen Programmfluss.

Empfehlung:

- Zu erwartende (oft fachliche) Fehler werden auf Rückgabewerte abgebildet.
- Unerwartete Fehler (oft technische Fehler, Logikfehler) werden über Exceptions behandelt.

Syntax für das Werfen und Fangen von Exceptions: `[exception_mit.php]`

Wichtig ist, dass Exceptions nicht bis zum Anwender durchschlagen dürfen, da sonst Implementierungsdetails preis gegeben werden und so das Programm angreifbar wird.

Ausnahmen, die das System wirft, führen zum Abbruch des Programms. Diese Fehler sind im Logfile des Apache (`/var/log/apache2/error.log`) zu finden. Sollen diese auch gefangen werden, so muss der Obertyp `Throwable` gefangen werden.

Aufgabe:

Ändern Sie das Beispiel `exception_mit.php` in folgender Weise ab:

- Entfernen Sie den `try/catch`-Block: Die Exceptions sind nun auf der Oberfläche sichtbar, falls das nicht mit `display_errors = Off` in der `php.ini` unterdrückt wird.
- Entfernen Sie die 0-Prüfung: Das Programm bricht ab (ab PHP 8).
- Ändern Sie den gefangenen Typ auf `Throwable`: Das Programm bricht nicht mehr ab.

Aufgabe:

Ergänzen Sie `add_medium.php`:

- In der Funktion `isSignatureInFile` soll für jede gelesene Zeile geprüft werden, ob es sechs Elemente sind (`count(...)`). Falls nein, soll eine Exception geworfen werden.
- Die Funktion `addMedium` soll einen `int`-Rückgabewert an den Aufrufer zurück geben. Definieren Sie dazu die Konstanten `RC_OK` und `RC_DUPLICATE`.
- Fügen Sie die entsprechenden `return`-Statements in `addMedium` ein.
- Werten Sie im Hauptteil von `add_medium.php` den Rückgabewert aus und fangen Sie die Exceptions.

3 Modularisierung

In „richtigen“ Anwendungen wird der Code auf mehrere Dateien verteilt. Zum einen sollen die Dateien nicht zu groß werden, zum anderen soll eine Codeverdoppelung vermieden werden.

PHP hat mit `require_once` eine Funktion, die es ermöglicht, externe Definitionen exakt einmal einzulesen. Ist eine externe Datei bereits eingelesen worden, wird die Anweisung beim zweiten mal ignoriert.

Für die Strukturierung des Codes und damit die Aufteilung in Dateien kann man auf bewährte Muster (*design patterns*) zurückgreifen. Das wichtigste Muster im Web-Umfeld ist das MVC-Muster (*model-view-control*). Dabei wird die Bearbeitung von HTTP-Anfragen nach folgenden Gesichtspunkten zerlegt:

- *model*: Alle Operationen, die mit den Daten (dem Datenmodell) zu tun haben. In unserem Beispiel wäre das das Lesen und Schreiben in die Medien-Datei.
- *view*: Alles, was mit der Anzeige, also mit der Erstellung der HTML-Ausgabe zu tun hat. Nur im *view* dürfen HTML-Tags vorkommen.
- *control*: Dieser Teil der Software nimmt die HTTP-Anfragen entgegen, leitet die extrahierten Daten an das *model* weiter und ruft zum Schluss den passenden *view*.

Das MVC-Muster gibt es in vielen Varianten. In Abb. 1 ist der Nachrichtenfluss einer sehr einfachen Variante gezeigt:

1. Von einem *view* aus wird eine neue Aktion angestoßen. Der Client wendet sich an den Server. Die Anfrage wird vom *controller* bearbeitet.

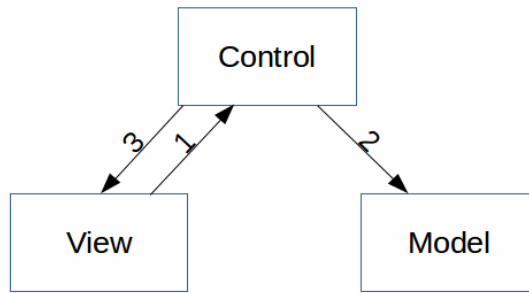


Abbildung 1: MVC-Muster

2. Dieser greift dazu auf das *model* zu. War es eine datenändernde Operation wird gemäß dem POST-REDIRECT-GET-Muster eine neue Anfrage an den *controler* ausgelöst.
3. Bei einer lesenden Operation werden die gelesenen Daten dem *view* übergeben. Dieser fügt die Daten in das HTML ein und schickt sie zurück an den Client.

Aufgabe:

Mit der folgenden Aufgabe soll der bereits bestehende Code in Module gemäß dem MVC-Muster gekennzeichnet werden.

- Entflechten Sie ggf. Ihre Dateien gemäß diesem Muster.
- Packen Sie alle *model*-Funktionen in eine Datei *Medienverwaltung.php*

4 Objektorientierung

4.1 Klassen

Klassendefinition: [oop_klasse.php (erster Teil)]

Anwenden einer Klasse: [oop_klasse.php (zweiter Teil)]

Konstruktor und Ausgabe: [oop_konstruktor_klassisch.php]

Constructor property promotion (PHP8): [oop_konstruktor.php]

Auf die klasseninternen Daten wird typischer Weise mit *getter* und *setter* zugegriffen:

```

function getGeschwindigkeit() {
    return $this->geschwindigkeit;
}
function setGeschwindigkeit($g) {
    $this->geschwindigkeit = $g;
}
  
```

In PHP werden Objektinstanzen ausschließlich im Freispeicher angelegt und vom *garbage collector* wieder frei gegeben. Eine Variable im Programm stellt daher lediglich einen Verweis in den Freispeicher dar. Über eine Zuweisung wird nur der Verweis kopiert, nicht die Instanz selbst. Um die Instanz selbst zu kopieren stellt PHP die Funktion `clone` zur Verfügung.

Soll beim Kopieren einer Instanz mehr passieren, als dass der Speicher, den die internen Daten der Instanz einnehmen, verdoppelt wird, kann hierfür eine Methode `__clone` angelegt werden. [`oop_kopie.php`]

Aufgabe:

Ergänzen Sie die Bibliotheksanwendung:

- Fügen Sie die Datei `add_user.html` hinzu, die Benutzerdaten (Name, Vorname, ...) abfragt und übermittelt (Vorlage: `add_medium.html`). `add_user.html` ruft beim *Submit* `add_user.php`.
- Legen Sie eine Datei `Benutzer.php` an. Definieren Sie darin eine Klasse `Benutzer` mit den entsprechenden Datenfeldern, einem Konstruktor, der die Felder versorgt, eine `__toString()`-Methode, die die Daten kommasepariert als Zeichenkette zurückgibt.
- Legen Sie eine Datei `Nutzerverwaltung.php` an, darin
 - eine Funktion `addUser`, die
 - * eine Instanz von `Benutzer` entgegennimmt,
 - * die Datei `/var/media/benutzer.csv` zum Anhängen öffnet,
 - * das Ergebnis der oben definierten `__toString()`-Methode an die Datei anhängt.
 - eine Funktion `getUser($id)`, die
 - * `/var/media/benutzer.csv` zum Lesen öffnet, die `$id`-te Zeile liest, eine `Benutzer`-Instanz erzeugt und diese zurückgibt.
 - * Falls es keine entsprechende Zeile gibt, wird `null` zurückgegeben.
- In `add_user.php`
 - werten Sie das Formular aus, legen mit diesen Daten eine Instanz von `Benutzer` an und rufen `addUser` auf.
 - Geben Sie die Instanz auf einer HTML-Seite aus.

Beziehungen

Objekte können Beziehungen zu anderen Objekten haben. Diese können unterschiedlich ausgestaltet werden. Entweder wird nur ein passender Schlüssel zum anderen Objekt gespeichert oder gleich eine Referenz auf das andere Objekt. Grundsätzlich haben beide Vorgehensweisen Vor- und Nachteile:

- Beim Anlegen geht es schneller, wenn vom referenzierten Objekt nur ein Schlüssel abgelegt werden muss.
- Beim Lesen geht es schneller, wenn das referenzierte Objekt bereits im Speicher vorliegt.

Da PHP keine getypte Sprache ist, können wir auch Objekte schreiben, die beides können: Im Konstruktor wird abgelegt, was übergeben wird: Schlüssel oder Objekt. Bei einem lesenden Zugriff wird geprüft, ob bereits eine Objektreferenz vorliegt. Falls nein, wird versucht über den Schlüssel ein entsprechendes Objekt zu erzeugen und zurückzugeben. Der Schlüssel kann dann durch die Objektreferenz ersetzt werden.

Aufgabe:

Schreiben Sie eine Klasse `Ausleihe`:

- Der Konstruktor hat als Parameter `Medium`, `Benutzer`.
- In der Methode `getNutzer` wird geprüft, ob es sich beim Wert der Membervariable um eine Instanz von `Benutzer` handelt. Falls nein, wird über `getNutzer` eine erzeugt und zurückgegeben.

Verwendet wird die Klasse erst später.

ClassLoader

In PHP-Programmen ist es praktisch, Klassen automatisch laden zu lassen. Dazu kann am System ein `classloader` angemeldet werden, der dann gerufen wird, wenn eine bis dato unbekannte Klasse verwendet werden soll. Dazu muss:

- Eine Klasse mit statischer Methode definiert werden. Die statische Methode bekommt den Klassennamen als Parameter.
- Die Methode sucht nach der Datei, die die gewünschte Klasse definiert und lädt sie mit `include`.
- Klassen- und Methodennamen werden mit `spl_autoload_register(array(CLASS_NAME, METHOD))` dem System bekannt gegeben.

Aufgabe:

In `add_user.php` mussten Sie die Klassendefinition explizit laden.

- Schreiben Sie eine Klasse `ClassLoader` mit der statischen Methode `autoload($class_name)`, die aus dem aktuellen Verzeichnis die Datei `"$class_name.php"` lädt.
- Laden Sie diese Klasse in `add_user.php`.
- Melden Sie Klasse und Methode zu Beginn von `add_user.php` an.
- `add_user.php` muss nun auch ohne das `include "Benutzer.php"` funktionieren.

Serialisierung

Um das POST/REDIRECT/GET-Muster implementieren zu können ist es oftmals nötig, den aktuellen Zustand einer Objektinstanz zwischenspeichern. Dazu dient `serialize` und `deserialize`. [T:282f, `oop_serial_*`]

Da sowohl die schreibende als auch die lesende Datei die Definition der Klasse `Fahrzeug` benötigt, wird die Definition der Klasse in eine eigene Datei ausgelagert und in beiden anderen Dateien mit `include` importiert.

Aufgabe:

Bringen Sie das Beispiel bestehend aus den Dateien `oop_serial_class.inc.php`, `oop_serial_put.php` und `oop_serial_get.php` zum Laufen.

Aufgabe:

Ergänzen Sie die Bibliotheksanwendung:

- Wenn in `add_user` das Benutzerobjekt erzeugt werden konnte, soll das Objekt serialisiert werden.
- Öffnen Sie eine Session und speichern Sie darin das Objekt.
- Leiten Sie nun auf die Seite `restore_user.php` weiter.
- `restore_user.php` deserialisiert die gespeicherte Instanz und gibt sie aus.

4.2 Vererbung

Der Vererbungsmechanismus wird häufig angewendet, um bereits vorhandene Definitionen zu übernehmen. Sie erzeugen dadurch eine Hierarchie von miteinander verwandten Klassen. Diese ermöglichen die Darstellung von Objekten, die teilweise übereinstimmende sowie auch unterschiedliche Merkmale aufweisen.

Im folgenden Beispiel wird eine Klasse `Pkw` definiert, mit deren Hilfe die Eigenschaften und Methoden von Personenkraftwagen dargestellt werden sollen. Bei der Erzeugung bedient man sich der existierenden Klasse `Fahrzeug`, in der ein Teil der gewünschten Eigenschaften und Methoden bereits vorhanden ist. Bei der Klasse `Pkw` kommen noch einige Merkmale hinzu. Hierbei handelt es sich um eine spezialisierte Klasse – im Gegensatz zur allgemeinen Klasse `Fahrzeug`.

Von der Klasse `Pkw` aus gesehen ist die Klasse `Fahrzeug` eine Basisklasse. Von der Klasse `Fahrzeug` aus gesehen ist die Klasse `Pkw` eine abgeleitete Klasse.³

³Theis: S. 270f.

Definition einer Vererbungshierarchie: [src/oop_vererbung.php, 1. Teil]

Sollen auch in der abgeleiteten Klasse die Elemente der Mutterklasse zugreifbar sein, so wird die Sichtbarkeit `protected` (statt `private`) gewählt.

Mit `parent::` kann auch der Konstruktor der Basisklasse wiederverwendet werden.

Instanziierung von Objekten: [src/oop_vererbung.php, 2. Teil]

Aufgabe:

Ergänzen Sie die Bibliotheksanwendung:

- Legen Sie die Datei `Medium.php` an. Darin
 - definieren Sie eine Klasse `MediumBase`, sowie die abgeleiteten Klassen `Buch` und `CD`.
 - Verteilen Sie die Attribute sinnvoll auf die Klassen. Das Typkennzeichen wird nicht mehr benötigt.
 - Die Konstruktoren von `Buch` und `CD` nehmen vier Parameter.
 - Implementieren Sie auch eine `__toString()`-Methode.
- Passen Sie den Classloader so an, dass wenn eine der implementierten Klassen gesucht ist, `Medium.php` geladen wird.
- Ergänzen Sie `Medienverwaltung.php` um die Funktion `getMedium($signatur)`, die für eine gegebene Signatur nach dem entsprechenden Medium sucht. Falls kein passendes Medium gefunden wird, wird `false` zurückgegeben (Vorlage: `isSignatureInFile`).
- Legen Sie die Datei `suche.php` an, die
 - ein Formular für *Signatur* und *Nutzernummer* enthält. Dieses Formular ruft beim *Submit* sich selbst auf (`suche.php` als `GET`).
 - falls in der Anfrage der Parameter `signatur` gesetzt und nicht leer ist, versucht über `getMedium` ein Medium zu laden und auszugeben.
 - falls in der Anfrage der Parameter `nutzer_id` gesetzt und nicht leer ist, versucht über `getNutzer` einen Benutzer zu laden und auszugeben.
 - falls beide Objekte geladen werden konnten
 - * eine `Ausleihe` instanziiert, diese serialisiert und speichert,
 - * der Ausgabe ein Formular anhängt, das einen `submit`-Knopf mit der Aufschrift *Ausleihen* hat und einen `POST`-Request auf `ausleihen.php` hervorruft.
- Legen Sie die Datei `ausleihen.php` an, die das Objekt deserialisiert und ausgibt.

4.3 Abstrakte Klassen, Schnittstellen und Traits

Da das Anlegen einer Instanz der Klasse `MediumBase` und auch die `format`-Methode mit den vielen Dummywerten keinen Sinn macht, kann so einem Fall eine *abstrakte Klasse* definiert werden.

Dazu wird mindestens eine Methode der Basisklasse als `abstract` gekennzeichnet:⁴

```
abstract class Animal
{
    public $name;
    public $age;

    public function Describe()
    {
        return $this->name . ", " . $this->age . " years old";
    }

    abstract public function Greet();
}

class Dog extends Animal
{
    public function Greet()
    {
        return "Woof!";
    }

    public function Describe()
    {
        return parent::Describe() . ", and I'm a dog!";
    }
}
```

Die Definition einer abstrakten Methode zwingt alle abgeleiteten Klassen diese zu implementieren.

Gibt eine Klasse nur „Implementierungsverpflichtungen“ in Form von abstrakten Methoden vor, nennt man das eine *Schnittstelle*. PHP unterstützt die Einfachvererbung, d.h. eine abgeleitete Klasse kann von einer Basisklasse erben, aber beliebig viele Schnittstellen implementieren.

Definition von Schnittstellen: [T:279, oop_schnittstelle.php]

Traits (Eigenschaft, Merkmal) bieten eine alternative Methode der Wiederverwendung: [src/traits.php]

5 Weitere Themen

5.1 Namensräume

Namensräume (englisch: *namespaces*) ermöglichen Ihnen, Elemente gleichen Namens zu definieren, zu nutzen und eindeutig zuzuordnen. So wie Sie Dateien gleichen Namens in unterschiedlichen Verzeichnissen auf Ihrer Festplatte haben dürfen, so dürfen Sie Klassen, Funktionen und Konstanten gleichen Namens in unterschiedlichen Namensräumen verwenden.⁵

⁴<http://www.php5-tutorial.com/classes/abstract-classes> (27.3.2018)

⁵Theis: S. 279

Definition und Verwendung: `[src/namespace.php]`

Vorsicht: Der derzeitige Interpreter akzeptiert (erste) Namensraumangaben nur am Anfang einer Datei. Dabei stören ihn vorgestellte BOM-Bytes⁶.

5.2 Reguläre Ausdrücke

In der Programmiersprache Perl sind Reguläre Ausdrücke fester Sprachbestandteil. In PHP sind sie über entsprechende Funktionen nutzbar. PHP implementiert die „Perl-kompatibel“ (PCRE: *Perl compatible regular expression*).

Regex-Tester⁷

PHP lässt für die Begrenzung des Ausdrucks auch andere Zeichen als `"/"` zu: Delimiters⁸

Einige Zeichen haben innerhalb regulärer Ausdrücke eine spezielle Bedeutung und müssen bei der Verwendung maskiert werden. Dies kann mit der Funktion `preg-quote`⁹ erfolgen.

<code>/</code>	leitet einen regulären Ausdruck ein
<code>.</code>	steht für ein beliebiges Zeichen
<code>+ * ? {}</code>	kennzeichnen Kardinalitäten
<code>^ \$</code>	verankern ein Suchmuster am Anfang/Ende
<code> </code>	kennzeichnet eine Auswahl
<code>()</code>	gruppiert Treffer
<code>[]</code>	definiert Buchstabenmengen

Suchmuster können mit Modifikatoren¹⁰ angepasst werden.

Wichtige PHP-Funktionen:

- Suche nach Treffern in einem String: `preg-match`¹¹
- Suche nach Treffer in einem Array: `preg-grep`¹²
- Suchen und Ersetzen: `preg-replace`¹³

Aufgabe:

Ergänzen Sie die Büchereianwendung um eine Parameterprüfung.

- Schreiben Sie in `Medienverwaltung.php` um eine Funktion `checkMedium()`, die die 6 Parameter übernimmt und mit regulären Ausdrücken prüft. Die Funktion soll `true` oder einen geeigneten Fehlertext zurückliefern.

⁶byte order mark

⁷<http://regex101.com> (21.2.2020)

⁸<https://www.php.net/manual/de/regexp.reference.delimiters.php> (12.11.2019)

⁹<https://www.php.net/manual/de/function.preg-quote.php> (12.11.2019)

¹⁰<https://www.php.net/manual/de/reference.pcre.pattern.modifiers.php> (12.11.2019)

¹¹<https://www.php.net/manual/en/function.preg-match.php> (12.11.2019)

¹²<https://www.php.net/manual/de/function.preg-grep.php> (12.11.2019)

¹³<https://www.php.net/manual/de/function.preg-replace.php> (12.11.2019)

- Rufen Sie die Funktion in `add_medium.php`. Falls die Funktion nicht `true` zurückliefert, rufen Sie über einen *redirect* den Errorview auf (`INVALID_PARAMETER`) und geben Sie den Fehlertext als weiteren Parameter mit.
- Ergänzen Sie den Errorview so, dass er bei Vorhandensein eines zusätzlichen Fehlertextes diesen auch anzeigt.

6 XML

PHP bietet zwei Implementierungen für den Umgang mit XML-Dateien:

- SimpleXML
- DOMDocument

Hier die Vor- und Nachteile¹⁴:

- SimpleXML
 - is for simple XML and/or simple UseCases
 - limited API to work with nodes (e.g. cannot program to an interface that much)
 - all nodes are of the same kind (element node is the same as attribute node)
 - nodes are magically accessible, e.g. `$root->foo->bar['attribute']`
- DOM
 - is for any XML UseCase you might have
 - is an implementation of the W3C DOM API (found implemented in many languages)
 - differentiates between various Node Types (more control)
 - much more verbose due to explicit API (can code to an interface)
 - can parse broken HTML
 - allows you to use PHP functions in XPath queries

6.1 SimpleXML

Lesen von XML-Dateien

Einlesen und auswerten eines XML-Dokuments: [`xml_sammlung.php`]

¹⁴<https://stackoverflow.com/questions/4803063/whats-the-difference-between-phps-dom-and-simplexml-extensions> (13.11.2019)

Schreiben von XML-Dateien

`addChild`¹⁵ Anmerkung: Das `include "example.php"` definiert die Variable `$xmlstr` mit dem Inhalt:

```
<?xml version="1.0" standalone="yes"?>
<movies/>
```

Aufgabe:

Erzeugen Sie ein XML-Dokument, das wie `xml_sammlung.xml` aussieht.

XPath

XPath¹⁶ ist der wahlfreie Zugriff auf XML-Knoten.

Aufgabe:

Extrahieren Sie aus `xml_mehrere.xml` alle Reifendaten.

6.2 DOMDocument

Das Arbeiten mit XML-Dokumenten über `DOMDocument` ist zwar für den Anfang etwas schwieriger, bietet aber Vorteile: Zum einen ist es standardisiert und kann in ähnlicher Weise auch in anderen Programmiersprachen implementiert werden. Zum anderen ist der Funktionsumfang größer.

Für den einfachen Umgang mit `DOMDocument` gibt es die Dateien `XmlExtractor.php` und `XmlFormatter.php`, deren Basisdefinitionen in `InputExtractor.php` und `OutputFormatter.php` zu finden sind.

Hier ein Beispiel zur Verwendung: `[src/xml_dom_appl.php]`

Aufgabe:

Erzeugen Sie ein XML-Dokument, das wie `ajax_xml_sammlung.xml` aussieht und lesen Sie es wieder ein.

Aufgabe:

Die Ausleihvorgänge sollen in einer XML-Datei abgelegt werden. Eine einzelne Ausleihe habe folgendes Format:

¹⁵<https://www.php.net/manual/de/simplexmlelement.addchild.php> (13.11.2019)

¹⁶<https://www.php.net/manual/de/simplexmlelement.xpath.php> (13.11.2019)

```
<ausleihen>
<ausleihe id="1">
    <from>04.05.2020</from>
    <to>04.06.2020</to>
    <medium signatur="A001"/>
    <user id="23"/>
</ausleihe>
</ausleihen>
```

- Legen Sie die Datei `Ausleiheverwaltung.php` an. Darin eine Klasse
 - die im Konstruktor die XML-Datei lädt, falls nicht vorhanden ein leeres XML anlegt,
 - `getAusleihe($signatur)`, die ein Ausleiheobjekt oder `false` zurückgibt,
 - `addAusleihe($ausleihe)`, die dem XML einen weiteren Ausleihe-Eintrag hinzufügt,
 - `save()`, die das XML wieder schreibt.
- Ergänzen Sie in `ausleihen.php` das Anlegen der Ausleihe. Prüfen Sie über `getAusleihe`, ob das Medium nicht schon verliehen ist. Vergessen Sie nicht am Ende mit `save()` die Änderung zu speichern.

Hinweis: Zum Formatieren von Datumsangaben beachten Sie die Datei `zeit_format.php`.

7 Ajax

Ajax steht für Asynchronous JavaScript and XML. Diese Technik bietet eine asynchrone Datenübertragung zwischen Browser und Webserver. Das ermöglicht Ihnen, Teile einer Internetseite zu ändern, ohne die gesamte Seite neu erstellen und übermitteln zu müssen. Auf diese Weise reduziert sich der Entwicklungsaufwand, verringert sich der Netzverkehr und werden die Internetseiten schneller und flexibler. Die Ajax-Technik wird seit Jahren erfolgreich im Web eingesetzt und ist ein selbstverständlicher Bestandteil vieler Internetseiten.¹⁷

Anmerkung: Im Firefox lässt sich mit F12 ein Javaskript-Debugger starten.

7.1 Hallo Ajax

Webseite, die über Ajax weitere Inhalte anfordert: [T:477, `ajax_hallo.htm`]

PHP-Programm, das die angeforderten Daten sendet: [T:478, `ajax_hallo.php`]

¹⁷Theis: S. 473.

7.2 Parameter senden

Im folgenden Beispiel soll nun ein Parameter (hier: MA-Id) mit an Server übergeben werden.

Webseite, die über Ajax weitere Inhalte zu einem bestimmten anfordert: [T:480, `ajax_parameter.htm`]

PHP-Programm, das die angeforderten Daten sendet: [T:481, `ajax_parameter.php`]

7.3 XML-Datei lesen

Bei den bisherigen Beispielen wurde nur Text eingefügt/ersetzt. Sollen Daten aus größeren Strukturen über Ajax in die bestehende Internetseite eingefügt werden, erfolgt dies über XML.

(Statisches) XML: [T:485, `ajax_xml_einzel.xml`]

HTML, das das XML anfordert und auswertet: [T:486, `ajax_xml_einzel.htm`]

Beispiel mit dynamischer Suche: [`src/suche.html/php`]

Aufgabe:

Nun wird die AJAX-Technik in das Projekt `Bucherei` integriert:

In `suche.php` sollen die Trefferkandidaten bei der Mediensuche angezeigt werden.

8 PHP 8

Like I said time and time again: PHP isn't the same language it was ten years ago, and we're very thankful for that. It's a fast and reliable language, used to build large applications at scale. So let's discuss some of the most notable changes of PHP in the last year, to the language and the community.¹⁸

Mit PHP 8 kamen folgende Neuerungen in die Sprache:

- PHP JIT

Ein *just in timer compiler* (JIT) compiliert den PHP-Code in die Maschinensprache des Zielsystems und speichert das Compilat in einem Zwischenspeicher. Wird dasselbe Modul ein zweites mal aufgerufen, kann direkt das Compilat aufgerufen werden, was deutlich performanter ist. Dies bringt dort Vorteile, wo wie einem Webserver oft derselbe Code auszuführen ist.

Um den JIT-Compiler zu aktivieren müssen in der `php.ini` (oder deren Unterdateien) folgende Einträge gemacht werden:¹⁹

¹⁸<https://stitcher.io/blog/php-in-2021>

¹⁹<https://stitcher.io/blog/php-8-jit-setup> (25.5.2021)

```
opcache.enable=1      # fur Server
opcache.cli_enable=1  # fur CLI
opcache.jit_buffer_size=100M
opcache.jit=1255      # max JIT mit tracing. S. Doku
```

Die Einstellungen in der `php.ini` können (wie immer) mit dem Aufruf `phpinfo()` überprüft werden.

- Constructor Property Promotion → 4.1
- Benannte Parameter: `[funktion_benannt.php]`
- Union Types → Union types²⁰
- Weitere Funktionen für Zeichenketten:
 - `str_contains(STR, "enthalten")`: Prüft, ob eine Teilzeichenkette enthalten ist
 - `str_starts_with(STR, "start")`: Prüft, ob STR mit "start" beginnt
 - `str_ends_with(STR, "end")`: Prüft, ob STR mit "end" endet
- Match Kontrollstruktur → Match or switch?²¹
- Nullsafe Operator → Einleitung²²
- Weak maps → The WeakMap class²³

Aufgabe:

- Stellen Sie die `switch`-Struktur in `error_view.php` auf `match` um.
- Dechiffrieren Sie das `return`-Statement in Weak maps²⁴ ("??" ist der Coalesce-Operator).

9 Quellen

- Theis, Thomas Einstieg in PHP7 und MySQL. Ideal für Programmierneinsteiger, 12. Auflage, 2017
- [T] Einstieg in PHP7 und MySQL. Ideal für Programmierneinsteiger, 13. Auflage, 2018

²⁰<https://php.watch/versions/8.0/union-types> (25.5.2021)

²¹<https://stitcher.io/blog/php-8-match-or-switch> (26.5.2021)

²²<https://stitcher.io/blog/php-8-nullsafe-operator> (26.5.2021)

²³<https://www.php.net/manual/en/class.weakmap.php> (26.5.2021)

²⁴<https://stitcher.io/blog/new-in-php-8> (26.5.2021)

Vielen Dank für Ihre Aufmerksamkeit

Ihr Referent und das Team von
IT-Schulungen.com

Fon +49 (0) 911 650 08 - 30
Fax +49 (0) 911 650 08 - 399
Mail info@it-schulungen.com
Web www.it-schulungen.com

Education Center der New Elements GmbH
Thurn-und-Taxis-Straße 10
90411 Nürnberg

www.newelements.de



New Elements GmbH | IT-
Schulungen.com