

# Technology Arts Sciences TH Köln



Konzeption und Erstellung einer  
internen Tauschbörse für  
Zeitschriftendoubletten in der  
Bibliothek des Zoologischen  
Forschungsmuseum Alexander  
Koenig Bonn

Abschlussarbeit für den Zertifikatskurs Data – Librarian

2021 Mareike Kruppa

## Inhalt

Einleitung.....	3
Motivation.....	4
Hintergrund .....	4
Ausblick .....	7
Umsetzung.....	8
Probleme und Erkenntnisse .....	12
Quellenverzeichnis .....	14

## Einleitung

Die Bibliothek des Zoologischen Forschungsmuseum Alexander Koenig in Bonn ist eine wissenschaftliche Spezialbibliothek für den Fachbereich der Zoologie. Die Nutzer der Bibliothek sind die Mitarbeiter der verschiedenen Wissenschaftlichen Sektionen des Hauses und die interessierte Fachöffentlichkeit.

Die allgemeinen, zoologischen und entomologischen Literaturbestände sind in der Zentralbibliothek und der Entomologischen Hauptbibliothek untergebracht. Jedoch verfügen die einzelnen Sektionen jeweils über eine eigene Bibliothek, in der sich spezielle Forschungsliteratur zu dem jeweiligen Fachgebiet befindet.

Auch diese Medien in den Sektionsbibliotheken werden von dem Personal der Hauptbibliothek gepflegt und verwaltet. Das Bibliothekspersonal besteht aus einem Diplombibliothekar und einer Fachangestellten für Medien- und Informationsdienste-Bibliothek.

Der Literaturbestand beläuft sich auf insgesamt ca. 217.000 Medieneinheiten, dabei handelt es sich um ca. 30.000 Monografien, 85.000 Sonderdrucke und rund 51.000 Zeitschriften.

Sammlungsschwerpunkte der Bibliothek ist Literatur der Wirbeltier- und Insekten der Paläarktis, Afrotropis und der Ornithologie des Paläarktis, Afrotropis und Neotropis.

Die Bibliotheken sind Präsenzbibliotheken und die Medien dürfen von externen Nutzern nur vor Ort eingesehen und kopiert werden.

## Motivation

Durch eine größere Anzahl Schenkungen und Nachlässe, von aktuellen und ehemaligen Wissenschaftlern sowie Citizen Science, ist der Platz für neue Medien in den letzten Jahren immer rarer geworden. Leider befinden sich in diesen Nachlässen und Schenkung eine große Anzahl an Dubletten. Vor allem im Bereich von Zeitschrifteneinzelheften und Sonderdrucken hat sich über die Jahre das Problem stark herauskristallisiert. Dies führt dazu, dass momentan keine Nachlässe und Schenkungen mehr angenommen werden können, wodurch der Bibliothek auch Chancen auf Erweiterung des Medienbestandes entgehen. Darüber hinaus führt die schiere Anzahl der Dubletten zu eklatantem Platzmangel sowie hohen Lagerhaltungskosten. Da wir diese Dubletten aus Kosten und Umweltgründen nicht entsorgen wollen, suchen wir schon seit längerem nach einer Lösung für dieses Problem.

## Hintergrund

Im Rahmen meiner Abschlussarbeit möchte ich mich dieses Problems annehmen und verfolge den Lösungsansatz eine „Tauschbörse“ in unserem Intranet zu schaffen.

Diese soll den Tausch von Sonderdrucken und Zeitschriften zwischen dem wissenschaftlichen Personal und der Bibliothek, aber auch zwischen den verschiedenen Wissenschaftlern unseres Institutes ermöglichen. Hier ergibt sich für die Bibliothek die Möglichkeit an weitere interessante Medien zu gelangen und unseren Bestand weiter zu vervollständigen. Gleichzeitig erhalten die Mitarbeiter die Gelegenheit an für sie interessante und neue Literatur zu gelangen, sowie ihren persönlichen Bestand an Literatur und insbesondere Dubletten zu verringern. Einen besonderen Wert legen wir auch auf eine Vermeidung der möglichen Umweltbelastung durch den Energieverbrauch bei weiterer Lagerung. Eine Vernichtung der Medien kommt aus Sicht der Bibliothek nicht in Frage, da es sich teilweise um alte, seltene und wissenschaftlich relevante Veröffentlichungen handelt. Für die Mitarbeiter bedeutet die Einführung der Tauschbörse zudem eine Arbeitserleichterung, da die manuelle Bearbeitung der Dublettenliste entfällt. Diese wird durch die Exportfunktion der Tauschbörse ersetzt, welche eine Tagesaktuelle Liste liefern können soll.

Die Daten für das Tauschprogramm werden aus einer bisher manuell gepflegten Exceltabelle übernommen. Diese Tabelle besteht aus fünf Spalten und ca. 800 Zeilen. Die Spalten sind mit Zeitschriftentitel, Herausgeber/Körperschaft, der ISSN, der Sprache sowie die Heftzählung angegeben. Die Zeilen sind alphabetisch nach dem Zeitschriftentitel sortiert. Es handelt sich hierbei um die

Zeitschriftendublettenliste die intern geführt wird und per E-Mail nach Bedarf dem wissenschaftlichen Personal des Zoologischen Forschungsmuseum Alexander Koenig zur Verfügung gestellt wird.

Die gewählten Mittel zur Umsetzung des Projektes sind an die Inhalte des Zertifikatskurses sowie der bereits im Museum vorhandenen Infrastruktur angepasst. Die Wahl fiel hier auf die Scriptsprache Python sowie WordPress als Frontend für die Website.

Die Python Scripts dienen der Umwandlung der bisherigen Excel Liste in ein Datenbankkompatibles CSV Format sowie dem Import und Export der Datenbankinhalte. WordPress stellt die Oberfläche der Website sowie die Grundlegenden Funktionen eines Webshops bereit, welchen ich anpassen werde, um eine Tauschbörse darzustellen. Zu den bereitgestellten Funktionalitäten zählen die Bestellfunktion, das Erstellen eines Warenkorbs sowie die Einbindung in die E-Mail-Infrastruktur unseres Instituts. Auf diese Weise sollte auch die Kommunikation zwischen einzelnen Benutzern sowie mit der Bibliothek sichergestellt sein.

WordPress ist bereits in einer angepassten Version innerhalb des Instituts im Einsatz. Es zeichnet sich meiner Meinung nach durch Benutzerfreundlichkeit aus. Dies macht es auch zu einem der beliebtesten Open Source Content Management Systemen im Bereich privater und kommerzieller Websites.

Im Rahmen der Projektumsetzung werde ich eine Standardversion von WordPress verwenden, da die angepasste Version des Museums nicht außerhalb des internen Netzes lauffähig wäre und auch nicht durch die IT-Abteilung für die Öffentlichkeit freigegeben ist.

Eine Portierung ist nach Rücksprache mit der IT-Abteilung aber problemlos möglich somit steht der Umsetzung meiner Meinung nach nichts im Wege.

Die Mindestanforderungen, welche die Tauschbörse erfüllen muss sind:

1. Eine Benutzerverwaltung mit Verifikation der E-Mail-Adresse (diese muss auf @leibniz-zfmk.de enden)
2. Die Möglichkeit, dass Medien sowohl manuell als auch in größeren Mengen automatisiert eingelesen werden können (beispielsweise als CSV Import)
3. Eine Suchfunktion integriert ist, welche eine einfache Identifikation der für den Benutzer relevanten Medien ermöglicht
4. Eine Bestandspflege innerhalb der Tauschbörse geschieht so dass jederzeit transparent ist welche Dubletten vorhanden sind

Durch die Einführung der Tauschbörse erhoffen wir uns eine bessere Übersicht über den vorhandenen Medienbestand. Sollte eine Medium nicht verfügbar sein so muss die Möglichkeit bestehen eine Beschaffungsanfrage an die Bibliothek zu stellen.

Durch die Zentralisierung dieses Prozesses erhoffen wir uns eine weitere Arbeitserleichterung für die Mitarbeiter der Bibliothek sowie eine Unterstützung bei der Bestandserweiterung durch die Nutzer. Des Weiteren wäre eine Möglichkeit der Kommunikation zwischen den Nutzern wünschenswert, so dass von Nutzern angebotene Medien leichter weiterverteilt werden können.

## Ausblick

Im Rahmen der Projektplanung habe ich mir auch Gedanken zu zukünftigen Erweiterungsmöglichkeiten gemacht. Hier sind mir insbesondere die folgenden Themen wichtig.

1. Es sollte für jeden Nutzer ersichtlich sein, ob ein Medium von der Bibliothek stammt oder von anderen Nutzern bzw. ob ein Medium für die Bibliothek oder andere Nutzer von Interesse ist. In der ersten Version werden wir nur Angebote unterstützen für die Zukunft wäre auch eine Möglichkeit Gesuche zu schalten wünschenswert.
2. Eine Auswertung der Suchanfragen wäre sinnvoll. So könnte die Bibliothek anhand dieser Daten Ableitungen treffen welche Medien für die Zukunft in welcher Stückzahl zu beschaffen sind.
3. Eine Erweiterung des Angebots, um andere Medientypen z.B. Monografien da perspektivisch auch hier ähnliche Probleme zu erwarten sind.
4. Eine Öffnung der Plattform nach außen bzw. zu ausgewählten Partnern z.B. innerhalb der Leibniz Gemeinschaft

Ich hoffe diese Verbesserungen in Kooperation mit unserer IT-Abteilung für die nächsten Jahre aufplanen und stückweise umsetzen zu können. Hier wurde bereits erstes Interesse signalisiert.

## Umsetzung

Die Umsetzung der mir gestellten Aufgabe begann mit ausführlicher Rechercharbeit zu Technologie und Methodologie einer Tauschbörse.

Grundlegende Funktionen wie die Benutzerverwaltung und die Darstellung von Websites konnte WordPress liefern aber eine Vorlage für eine Tauschbörse wie sie mir vorschwebte gab es nicht.

Ich verfolgte verschiedene Ansätze.

1. Komplette Neuentwicklung eines WordPress Themes inkl. der darunterliegenden Datenbank Struktur
2. Ein Plugin verwenden, welches die Funktionalität einer Tauschbörse abbildet
3. Ein Plugin finden welches mir die Grundfunktionen wie die Datenbank und Warenkorb-Funktionen bereitstellt, jedoch Customizing erfordert, um meine Anforderungen zu erfüllen

Die Recherche zeigte recht früh, dass es kein frei verfügbares Plugin zu geben scheint, welches die von mir gestellten Anforderungen an das Projekt vollumfänglich erfüllt. Es gibt zwar Tauschbörsen/Kleinanzeigen Plugins wie z.B. „classified-listings“ jedoch waren diese nicht mit den Funktionalitäten ausgestattet welche ich in meinen Anforderungen als notwendig festgelegt hatte<sup>1</sup>.

Eine komplette Neuentwicklung war mein nächster Ansatz. Allerdings stellte ich recht schnell fest, dass dies bei meinem aktuellen Kenntnisstand sehr viel mehr Zeit in Anspruch nehmen würde als mir zur Verfügung stand. Allein meine ersten Recherchen zum Thema Word Press Themes erstellen und PHP im Allgemeinen ließen meine Einschätzung der Projektzeit auf weit über 50 Stunden steigen.

Der Dritte Ansatz zur Lösung meines Problems sollte sich als erfolgreich herausstellen. Ich suchte nach einem Plugin, welches mir Funktionalitäten eines Online-Shops liefert. Meine Idee war, dass meine Tauschbörse durchaus einem Shop wie Amazon ähneln könnte. Verschiedene Anbieter können dort Ihre Produkte anbieten und Ihrem Kundenkreis zugänglich machen. Der Unterschied wäre, dass alle meine „Produkte“ kostenlos angeboten würden und der Kundenkreis auf die Mitarbeiter der Zoologischen Museums Alexander Königs beschränkt sein sollte.

Meine Recherchen führten mich nach einiger Zeit zu dem WordPress Plugin WooCommerce. Dieses bietet die grundlegenden Funktionen für eine Multivendor Plattform, das heißt eine Plattform auf verschiedene Anbieter Produkte und Dienstleistungen anbieten und verkaufen können. Also genau meiner Vorstellung entsprach. Darüber hinaus ist auch dieses Plugin Open Source und sowohl die Bedienung per GUI<sup>2</sup> als auch die REST-API<sup>3</sup> sehr gut dokumentiert. Des Weiteren gibt es eine von den Entwicklern bereitgestellte Python Library für die Kommunikation mit der API.



Dies schien mir eine solide Grundlage für meine Tauschbörse zu sein, allerdings wusste ich auch das ein gewisses Maß an Individualisierung nötig sein würde.

Ich musste mich in die Grundlagen von PHP und der Funktionsweise von WordPress Themes einarbeiten, um die Individualisierungen umzusetzen. Hier kam mir meine Recherche zur Neuentwicklung zugute. Hier hatte ich bereits einige Quellen durchgearbeitet. Besonders positiv blieb mir hier das ausführliche Tutorial von lernhoch2<sup>4</sup> in Erinnerung. Hier habe ich regelmäßig nachgeschlagen, um mir die Strukturen in PHP wieder in Erinnerung zu rufen.

Nachdem meine Technologieauswahl abgeschlossen war, begann ich also die Umsetzung meiner Anforderungen in der gewählten Technologie.

Im Rahmen dieser Abschlussarbeit verwende ich eine Virtuelle Maschine von bitnami.com mit vorinstallierter WordPress Basis Instanz<sup>5</sup>. Auf diese Weise kann mein Ergebnis mit einem beliebigen Hypervisor repliziert werden. In unserem Intranet würde die WordPress Instanz von unserer IT-Abteilung bereitgestellt werden.

Die Einrichtung der VM gemäß der Dokumentation verlief problemlos<sup>6</sup>. Nach dem Start führte ich als erstes ein Update über die GUI aus. Somit war meine Umgebung bereit und die Entwicklung konnte beginnen.

Ich installierte das WooCommerce Plugin über die WordPress Admin GUI und durchlief die geführte Ersteinrichtung des Onlineshops. Hier fiel mir der erste Individualisierungsbedarf auf. Standardmäßig ist ein Versand der Waren vorgesehen, weshalb Nutzer zwingend eine Versandadresse angeben müssen. Dies passte nicht zu meinem Tauschbörsenkonzept, welches nur eine Abholung in der Bibliothek vorsah.

Nachdem die Grundinstanz des Onlineshops lauffähig war begann ich mit der Entwicklung meiner Python Skripte. Hier wollte ich 3 Hauptfunktionen abbilden.

1. Die bestehende Excel Datei umformatieren und in ein API kompatibles Format überführen
2. Die in 1 erzeugte Datei über die API an die Tauschbörse übermitteln, um den Warenbestand so aktuell halten zu können
3. Eine Abfrage des aktuellen Warenbestands auszuführen und daraus eine CSV Datei zu erzeugen, um diese in Excel verfügbar zu machen

Die Feldnamen der API sind in der Dokumentation in den Code Beispielen zu finden also machte ich mich daran eine Kreuz-Referenzliste zu erstellen. Nachdem dies erfolgt war, exportierte ich die Excel Datei als CSV und begann mit der Programmierung des CSV-Formatierer Skripts. Hier muss sich die Datei nur im gleichen Stramm Verzeichnis befinden bzw. ein absoluter Dateipfad eingegeben werden.

Nachdem dies erfolgreich abgeschlossen war musste ich nun die Erstellten „Produkte“ in die Tauschbörse hochladen. Hierzu bietet WooCommerce in der GUI

zwar eine Importfunktion, ich wollte dies aber auch Scriptgestützt umsetzen um ggf. in Zukunft neue Listen automatisiert einlesen zu können.

Um mit der API kommunizieren zu können sind entsprechende API Keys notwendig. Die Erzeugung dieser Keys gemäß Dokumentation verlief problemlos<sup>7</sup>. Nachdem ich die Keys erzeugt hatte, begann ich damit die erstellte CSV Datei in JSON kompatible Daten umzuformatieren und über die API zu posten.

Hier stieß ich auf einige Probleme. So wurden alle Produkte im ersten Moment als Unsichtbar hochgeladen da ich den Wert für „published“ nicht auf 1 gesetzt hatte. Einige Iterationen des Skripts später waren diese Fehler jedoch behoben und das automatisierte Update der Medien funktionierte.

Die letzte Funktionalität, welche ich über ein Python Skript darstellen wollte, war der Export der Bestandsdatenbank. Auch hier bezog ich mich auf die API-Dokumentation. Es gibt hier eine Funktion für „List All Products“ jedoch liefert diese im Standard 10 und maximal 100 Ergebnisse zurück. Hier werden die Ergebnisse auf Pages aufgeteilt. Mein Skript musste also alle Pages lesen und die Daten in einer CSV zusammenführen. Auch dies gelang mir nach einigen Versuchen.

Somit waren meine 3 Technischen Ziele erfüllt und ich konnte mich dem Customizing meines Shops widmen.

Folgende Funktionen wollte ich ändern bzw. schaffen:

1. Wie zuvor bereits erwähnt geht das System von Versand als Standardmethode aus und verlangt eine Versandadresse. Dies sind Daten, die ich nicht erheben möchte und auch keinen Anlass dazu habe
2. Preise sind in meinem Anwendungsfall nicht notwendig, da alle Angebote auf der Plattform kostenlos sind
3. Da es sich bei WooCommerce um ein Plugin für Shops handelt sind die Funktionen mit „zur Kasse gehen“ und „Kostenpflichtig bestellen“ beschrieben. Ich möchte aber eine Kostenlose Reservierung anbieten
4. Es soll nur Nutzern mit einer ZFMK E-Mail-Adresse möglich sein sich zu registrieren.

Alle diese Probleme sind mit Konfigurationen bzw. Veränderungen des WordPress-Themes verbunden.

Für das erste Problem habe ich recht schnell eine Lösung gefunden. Tatsächlich ist es möglich in den erweiterten Einstellungen von WooCommerce den Zwang einer Versandadresse aufzuheben, sowie die entsprechenden Felder aus dem Bestellformular zu entfernen. Hier habe ich nur den Vor- und Nachnamen erhalten, um eine Zuordnung der Bestellungen zu ermöglichen.

Die anderen Probleme konnten nicht so einfach gelöst werden. Hier waren Eingriffe in das PHP des WordPress Themes nötig. Nach einiger Recherche wurde mir klar, dass ich über Funktionen in der „functions.php“ Datei die gewünschten Ergebnisse erzielen kann<sup>8</sup>.

Anhand der Dokumentation der Action and Filter Hooks in WooCommerce habe ich mich an den entsprechenden Filtern versucht. Nach einigen Versuchen, welche mal mehr mal weniger erfolgreich verliefen, konnte ich die richtigen Hooks identifizieren und die Funktionen entsprechend beschreiben.

Vor eine besondere Herausforderung stellte mich hierbei die Anforderung der Nutzerbeschränkung. Das Problem war den korrekten Hook zu finden, welcher noch vor Anlage des Nutzers mögliche Fehler abprüft. Nachdem ich diesen jedoch identifiziert hatte, war das Umsetzen der Funktion in einem angemessenen Zeitraum zu bewältigen.

Das Ergebnis ist die lauffähige Umgebung, welche ich im Rahmen dieser Abschlussarbeit eingereicht habe.

---

## Probleme und Erkenntnisse

Während der Umsetzungsphase gab es verschiedene Entwicklungen. Sowohl positive wie negative, die wichtigsten möchte ich im Folgenden kurz beschreiben und meinen Umgang mit ihnen darlegen.

Positiv ist hervorzuheben das sowohl WordPress als auch die bereits im Zertifikatskurs verwendete Python Umgebung sehr gut funktioniert haben. Die Anforderungen an Benutzerfreundlichkeit wurden auch für einen Anfänger wie mich vollumfänglich erfüllt.

Die Installation von WordPress war dank einer vom Hersteller vorgefertigten virtuellen Maschine innerhalb weniger Minuten bereit, so dass ich mit der Einrichtung der Website beginnen konnte. Die Dokumentation ermöglichte mir ein schnelles Verständnis der Funktionsweise und Einstellungsparameter, welche für die Umsetzung meines Projekts relevant sein würden.

Das Wissen zur Programmierung in Python, welches im Kurs erlangt wurde, ermöglichte mir zwar die Umsetzung der Grundfunktionen, wie z.B. die Überführung der alten Excel Liste in ein importierbares CSV Format. Jedoch wurden für einige Funktionen wie z.B. den automatisierten Upload von größeren Medienlisten im Kurs nicht behandelt. Hier war ein gewisser Rechercheaufwand nötig, jedoch möchte ich auch hier auf die hervorragende Dokumentation des Python sowie des WordPress Projektes verweisen. Diese waren stets hilfreich und haben mir das nötige Wissen vermittelt.

Dieser Informationsmangel führte jedoch zu einem erhöhten Aufwand für die technische Umsetzung. Das Programm durchlief 11 Versionen, bevor es fehlerfrei die ihm zugedachte Funktion erfüllte. Dieser erhöhte Zeitaufwand bedeutete eine Verzögerung in meinem Projektplan von einigen Tagen. Dies relativierte sich jedoch, da ich Großteile des Import Programms für den Export wiederverwenden konnte, da die API hier nahezu identisch ist.

Somit konnte ich mein Projekt im geplanten Zeitrahmen lauffähig machen und meine selbstgesteckten Meilensteine erreichen.

Als ich meinem Arbeitgeber eine erste Version des Projekts vorstellte bestanden Zweifel bezüglich der Umsetzbarkeit, der Anwenderfreundlichkeit sowie des allgemeinen Nutzens für die Bibliothek. Mit fortschreiten des Projekts wurde das Interesse und Vertrauen meines Arbeitgebers in die Tauschbörse jedoch immer größer. Inzwischen ist es Bestandteil des Weiterentwicklungskonzept der Bibliothek. Die von mir aufgeführten Probleme des Platzmangels werden besonders vor dem Hintergrund eines anstehenden Umzuges sowie des im allgemeinen gestiegenen Interesse am Thema Nachhaltigkeit als systemrelevant definiert und mein Projekt als Lösungsweg anerkannt.

Sollte die Tauschbörse ein Erfolg werden und wir den Dublettenbestand erfolgreich reduzieren, so können wir in naher Zukunft wieder Schenkungen und Nachlässe annehmen, sowie gezielt nach Desideraten in beliebten Reihen suchen, um die Qualität unserer Bibliothek weiter zu verbessern.

---

## Quellenverzeichnis

- 1) <https://de.wordpress.org/plugins/classified-listing/>
- 2) <https://docs.woocommerce.com/documentation/plugins/woocommerce/>
- 3) <https://woocommerce.github.io/woocommerce-rest-api-docs/#introduction>
- 4) <http://wordpress.lernenhoch2.de/handbuch/eigenes-wordpress-template-erstellen/>
- 5) <https://bitnami.com/stack/wordpress/virtual-machine>
- 6) <https://docs.bitnami.com/virtual-machine/apps/wordpress>
- 7) <https://woocommerce.github.io/woocommerce-rest-api-docs/?python#authentication>
- 8) <https://docs.woocommerce.com/document/introduction-to-hooks-actions-and-filters/>