

Anhang A

Anleitung

Eine kurze Schritt-für-Schritt Anleitung für die Inbetriebnahme der Anwendung.

Damit der Dockercontainer der Anwendung benutzt werden kann, muss zuerst ein Dockerimage gebaut und in eine Dockerregistry gepusht werden. Dafür wird der Anwendungscode benötigt. Dieser befindet sich auf der beigelegten CD.

A.1 Dockerimage erstellen und hochladen

1. CD einlegen und „learnWebApps.zip“ abspeichern.
2. In den Unterordner „learnWebApps“ wechseln.
3. Datei „docker-compose-prod.yml“ öffnen.
4. In dem „backend“ und „frontend“ Service das „image“ Feld ändern. Dort wird der Name des zu erstellenden Images nach dem Muster **[registry]/[projekt]:[tag]** eingetragen.
5. Folgenden Befehl im Terminal ausführen, um das Image zu bauen:

```
docker-compose -f docker-compose-prod.yml build
```

6. Sobald das Image gebaut ist, können die Images in die registry gepusht werden. Dazu wird folgender Befehl verwendet:

```
docker-compose -f docker-compose-prod.yml push
```

A.2 Anwendung starten

1. Auf dem Server einloggen, auf dem die Anwendung laufen soll.
2. Die Datei „learnWebApps.yml“ aus dem Repository kopieren und auf den Server legen.
3. In der Datei „learnWebApps.yml“ die „image“ Felder anpassen.
4. Eine Datei namens „.env“, die wichtige Umgebungsvariablen enthalten soll, anlegen.

5. Folgende Umgebungsvariablen nach dem Muster **Name=Wert** in die Datei „.env“ schreiben:

- DB_CONTAINER_NAME
- SQL_USER
- SQL_PW
- DATABASE_NAME
- SESSION_KEY
- FRONTEND_URL (mit prefix http:// oder https:// und Port Angabe)
- BACKEND_URL (mit prefix http:// oder https:// und Port Angabe)

6. Die Anwendungscontainer können nun mit folgendem Befehl gebaut und gestartet werden:

```
docker-compose -f learnWebApps.yml --env-file .env up
```

7. Nachdem die Container gestartet sind, muss ein Backendaccount für den Dozenten erstellt werden. Dies kann mit folgendem Befehl gemacht werden. Nach dem Ausführen wird nach einem Namen und einem Passwort gefragt, mit denen sich der Dozent ins Backend einloggen kann.

```
docker-compose exec backend node createAdmin.js
```

8. Nun kann sich der Dozent einloggen und Lerninhalte einpflegen.

A.3 Einpflegen von Lerninhalten

1. Hinweise auf der Startseite beachten.
2. Über die Kachel mit dem Plus-Icon (+) können neue Kapitel, Unterthemen, Informationseinheiten und Aufgaben erstellt werden.
3. Mithilfe der Kachel mit dem Stift-Icon (✎) können Kapitel, Unterthemen, Informationseinheiten und Aufgaben bearbeitet werden.
4. Über die Kachel mit folgendem Icon (📎) können Mediendateien hochgeladen werden. Es werden gif, png, jpg, jpeg und mp3 Dateien akzeptiert. Wie diese eingebunden werden, wird beim Erstellen und Bearbeiten von Aufgaben angezeigt.
5. **Kapitel** bestehen aus folgenden Feldern:
 - Kapitelname
 - Übersichtstext (kann HTML enthalten)
 - Kapitelnummer (spiegelt die Reihenfolge wieder, in der die Kapitel angezeigt werden)
6. **Unterthemen** bestehen aus folgenden Feldern:

- Unterthemenname
- Übersichtstext (kann HTML enthalten)
- Unterthemennummer (spiegelt die Reihenfolge wieder, in der die Untertemen angezeigt werden)
- Kapitel (zeigt an, zu welchem Kapitel das Unterthema gehört)

7. **Informationseinheiten** bestehen aus folgenden Feldern:

- Aufgabenname
- Informationen (können HTML enthalten)
- Aufgabennummer (spiegelt die Reihenfolge wieder, in der die Informationseinheiten / Aufgaben angezeigt werden)
- Unterthema (zeigt an, zu welchem Unterthema die Informationseinheit / Aufgabe gehört)

8. **Single- und Multiple-Choice-Aufgaben** bestehen aus einer Informationseinheit und folgenden Feldern:

- Schwierigkeitsgrad (alle, einfach, mittel, schwierig)
- Hinweise / Tipps (optional)
- Rückmeldung nach dem erfolgreichen Lösen (optional)
- Mögliche Antworten (Richtige Lösungen werden mit einem [X] markiert. Eine mögliche Lösung wird in eine Zeile des Textfeldes geschrieben.)

9. **Codeaufgaben** bestehen aus einer Informationseinheit und folgenden Feldern:

- Schwierigkeitsgrad (alle, einfach, mittel, schwierig)
- Hinweise / Tipps (optional)
- Rückmeldung nach dem erfolgreichen Lösen (optional)
- Überprüfungstyp (Jasmine oder Self; entscheidet darüber, ob Jasmine zum Überprüfen des Nutzercodes benutzt werden soll oder nicht)
- Musterlösung
- Überprüfungsinformationen (können HTML enthalten; sollen genau beschreiben, was getestet wird)
- Überprüfungscode (Code zum Überprüfen des Nutzercodes)
 - Bei jeder Überprüfungs-methode ist die Variable „codeFromUser“ vorhanden. Diese ist ein String und beinhaltet den Code vom Nutzer.
 - Wird Jasmine nicht verwendet, ist die Variable „errorMessagesForUser“ vorhanden. Diese ist ein Array von Strings. In sie sollen Fehlermeldungen (Strings) gepusht werden, die dem Nutzer angezeigt werden.

10. **Codeergänzungsaufgaben** bestehen aus einer Informationseinheit und folgenden Feldern:

- Schwierigkeitsgrad (alle, einfach, mittel, schwierig)
- Hinweise / Tipps (optional)
- Rückmeldung nach dem erfolgreichen Lösen (optional)
- Zu ergänzender Code mit [input] als Lücken (kann HTML enthalten; Lücken, die der Nutzer füllen soll, werden mit [input] markiert.)
- Antworten (Jede Lösung für ein Input bekommt im Textfeld eine eigene Zeile. Die Reihenfolge der Lösungen entspricht der Reihenfolge der Inputs.)

11. **Lückentextaufgaben** bestehen aus einer Informationseinheit und folgenden Feldern:

- Schwierigkeitsgrad (alle, einfach, mittel, schwierig)
- Hinweise / Tipps (optional)
- Rückmeldung nach dem erfolgreichen Lösen (optional)
- Zu ergänzender Text mit [input] als Lücken (Lücken, die der Nutzer füllen soll, werden mit [input] markiert.)
- Mögliche Antworten (In dem Textfeld soll pro Zeile nur eine mögliche Lösung angegeben werden. Richtige Lösungen werden mit einem [X] markiert. Die Reihenfolge der mit [X] markierten Lösungen entspricht der Reihenfolge der Inputs.)

A.4 Einbinden in Moodle

1. Moodle Seite öffnen und anmelden.
2. Auf folgende Unterseite wechseln: „Website-Administration > Plugins > Aktivitäts Module > Externes Tools > Tool verwalten“.
3. In das Inputfeld mit dem Platzhalter „Tool-URL...“ folgendes eintragen und anschließend auf den grünen Button „LTI Advantage hinzufügen“ klicken.
 - BACKEND_URL + /register
4. Bei folgender Box (s. Abb. **A.1**), die nach einer kurzen Ladezeit auftaucht, ebenfalls auf den grünen Button mit der Aufschrift „Aktivieren“ klicken.

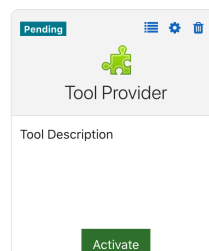


Abbildung A.1: Lernanwendung aktivieren
Quelle: Screenshot von Moodle

5. Damit sich die Anwendung in einem eigenen Browserfenster öffnet, in der Box (s. Abb. A.1) auf das Zahnrad (Einstellungen) klicken.
6. Nachdem sich die Einstellungen geöffnet haben, unter „Standard-Startcontainer“ die Möglichkeit „Neues Fenster“ auswählen und speichern (s. Abb. A.2).

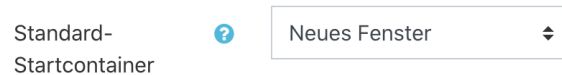


Abbildung A.2: Lernanwendung in einem neuen Fenster
öffnen

Quelle: Screenshot von Moodle

7. Um die Anwendung in einen Kurs zu integrieren, in dem Kurs auf den Link „Aktivität oder Material anlegen“ klicken und anschließend in dem Pop-Up die Möglichkeit „Externes Tool“ auswählen.
8. Auf der erscheinenden Seite unter „Vorkonfiguriertes Tool“ die jeweilige Anwendung auswählen und speichern.