Contents

[Einrichten einer Moodle-Instanz 2](#_Toc138421276)

[Installation von Plugins 6](#_Toc138421277)

[Code Plugin 7](#_Toc138421278)

[version.php 7](#_Toc138421279)

[question.php 8](#_Toc138421280)

[questiontype.php 8](#_Toc138421281)

[edit\_code\_form.php 8](#_Toc138421282)

[renderer.php 8](#_Toc138421283)

[DB 8](#_Toc138421284)

[Lokalisierung 10](#_Toc138421285)

[Workflow: Parameter hinzufügen 10](#_Toc138421286)

[Monaco-Editor 11](#_Toc138421287)

[Language Server 12](#_Toc138421288)

[Links 12](#_Toc138421289)

# Einrichten einer Moodle-Instanz

Um Moodle zu hosten ist ein Web- und Datenbankserver notwendig. Da Moodle auf PHP basiert ist ein simpler Webserver wie Apache ausreichend. Als Datenbank ist MySQL oder MariaDB zu empfehlen. Ein einfacher Weg eine Entwicklungsumgebung aufzusetzen ist XAMMP: <https://www.apachefriends.org/>

Moodle verlangt folgende Änderungen in der php.ini

max\_input\_vars = 50000

max\_execution\_time = 120

post\_max\_size = 256M

upload\_max\_filesize = 256M

Die php.ini kann unter XAMMP wie folgt geöffnet werden:

A screenshot of a computer

Description automatically generated

Nach der Installation von XAMMP muss das Moodle-Repository geklont werden (<https://github.com/moodle/moodle>). Ein git clone Kommando, welches eine Stable-Branch klont, ist: git clone --branch MOODLE\_401\_STABLE git://git.moodle.org/moodle.git

Für eine Stabile Entwicklungsumgebung ist die kombination aus Moodle\_401 und XAMMP 7.4.33 empfohlen (<https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/7.4.33/> ).

Der Moodle-Ordner ist entweder in den Ordner xampp/htdocs zu legen, oder ein Link ist zwischen dem Moodle-Ordner und dem htdocs-Ordner zu erstellen. Nach dem Starten des Apache- und des DB-Servers kann die Moodle-Installation mit dem Besuchen von localhost/moodle begonnen werden.

A screenshot of a computer

Description automatically generated

XAMPP verwendet MariaDB. Unter XAMPP sieht die Datenbankeinrichtung so aus:

A screenshot of a computer

Description automatically generated with medium confidence

Etwaige Probleme werden auf der folgenden Seite angezeigt. Wenn Plugins fehlen, sind die entsprechenden Zeilen in der php.ini auszukommentieren.

A screenshot of a computer

Description automatically generated with medium confidence

Danach führt Moodle die Erstellung und Initialisierung der Datenbank durch, dass kann einige Minuten dauern. Danach sind einige abschließende Einstellungen vorzunehmen.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Danach ist Moodle installiert und kann verwendet werden.

A screenshot of a computer

Description automatically generated with medium confidence

## Installation von Plugins

Plugins können mittels „Site administration“ -> „Plugins“ -> „Install plugins“ installiert werden. Dabei werden sie gezippt in das Feld gezogen. Moodle platziert dann automatisch das Plugin im entsprechenden Ordner.

A screenshot of a computer

Description automatically generated with medium confidence

# Code Plugin

Die Grundlagen der Plugin-Entwicklung sind nicht teil dieses Dokuments. Tutorials und „Kurse“ sind in unter Moodle-Acadamy zu finden. Für dieses Plugin sind die Moodle-Developer Kurse am relevantesten. <https://moodle.academy/>

A screenshot of a website

Description automatically generated with low confidence

Das Moodle-Plugin besteht folgenden Dateien:

* version.php
* question.php
* questiontype.php
* edit\_code\_form.php
* renderer.php
* DB-Dateien
* Lokalisierung

## version.php

In dieser Datei wird die Version des Plugins, im Format YYYYMMDDXX angeben. Moodle berücksichtigt erst änderungen in der Datenbank, wenn die Versionsnummer erhöht wird.

## question.php

Die Datei question.php definiert die Instanz einer Frage. Dabei werden die verschiedenen fragenspezifischen Konfigurationsparameter als Variablen definiert.

## questiontype.php

Hier werden Metainformationen über den Typ gespeichert. Dabei kann für bestimmte Parameter defineiert werden welche Werte sie annehmen können. Es kann auch für Parameter ein Default-Wert gesetzt werden. Weiters sind hier die Funktionen zum Speichern einer Frageninstanz und zum Erstellen einer solchen Instanz.

## edit\_code\_form.php

Diese Datei beschreibt die Eingabemaske der definierten Parameter einer Frage. Dabei wird mittels der Moodle-Forms-API ein Formular erstellt. <https://moodledev.io/docs/apis/subsystems/form>

## renderer.php

Hier wird die Frage während eines Tests dargestellt. Dabei werden die relevanten Parameter in unsichtbaren p-Tags geschrieben. Die Initialisierung des Monaco-Editors geschieht mit der Ausgabe eines script-Tags. Dieses script-Tag zeigt auf die Datei app.js im Dist-Ordner des Plugins.

## DB

In Moodle sind die Tabellen eines Plugins in der Datei install.xml definiert. Entwickler müssen diese Datei nicht per Hand verändern, sondern können Moodle-Dev-Tools benutzen:

Site administration 🡪 XMLDB editor

A screenshot of a computer

Description automatically generated with medium confidence

Danach das Plugin suchen. In diesem Fall ist es: question/type/code

A screenshot of a computer

Description automatically generated with low confidence

Danach: Load + Edit

A screenshot of a computer

Description automatically generated with medium confidence

Nun könne die Tabellen des Plugins bearbeitet werden. Nachdem die Änderungen in install.xml gespeichert wurden, muss die Datei upgrade.php erweitert werden. Unter dem Punkt „View PHP code“ kann dieser Upgrade-Code generiert werden. Dieser muss dann nur noch in die Datei kopiert werden. Damit Moodle diese Änderung erkennt, muss die Versionsnummer in der Datei version.php angehoben werden.

A screenshot of a computer

Description automatically generated with medium confidence

## Lokalisierung

In Moodle-Code kann mittels get\_string('ID', 'qtype\_code') auf lokalisierte Strings zugegriffen werden. Diese Strings befinden sich in der qtype\_code.php Datei. Für jede unterstützte Sprache gibt es eine solche Datei. Der Inhalt einer dieser Dateien sieht so aus: $string['answertext'] = 'Answer text';

## Workflow: Parameter hinzufügen

Um einen Parameter zu der Code-Frage hinzuzufügen, dabei müssen folgende Änderungen vorgenommen werden:

* question.php

In der Datei question.php muss der gewünschte Parameter als Variable hinzugefügt werden.

* questiontype.php

In questiontype.php müssen die Funktionen save\_question\_options und initialise\_question\_instance erweitert werden. Die Erweiterung ist in der Funktion save\_question\_options als $options->newVar = initialValue und $options->newVar = $formdata->newVar durchzuführen. Die Funktion initialise\_question\_instance ist mit $question->newVar = $questiondata->options->newVar zu erweitern.

* edit\_code\_form.php

Hier sind die Funktionen definition\_inner und data\_preprocessing zu erweitern. In definition\_inner ist, mithilfe der Form-API, die Eingabemaske zu erweitern. In data\_preprocessing ist $question->newVar = $question->options->newVar hinzuzufügen.

* install.xml

Diese Datei wird mittels des XMLDB-Editors erweitert.

* upgrade.php

Diese Datei muss mit dem PHP-Code vom XMLDB-Editor erweitert werden.

* renderer.php

Hier werden die HTML-Tags für den Monaco-Editor generiert. In der Funktion formulation\_and\_controls kann mittels $qa-> get\_question()->newVar auf den neuen Parameter zugegriffen werden.

* version.php

Wie schon erwähnt, muss die Version erhöht werden damit Moodle die Datenbank aktualisiert.

# Monaco-Editor

Der Monaco-Editor ist mithilfe einer separaten Datei eingebunden. Um einen neuen LS einzubinden, muss folgendes IF-Konstrukt erweitert werden:

if (language == 'java') {

var ext = '.java';

} else if (language == 'csharp') {

var ext = '.cs';

} else if (language == 'pascal') {

var ext = '.pas';

} else if (language == 'python') {

var ext = '.py';

} else if (language == 'c'){

var ext = '.c';

} else if (language == 'cpp') {

var ext = '.cpp'

}

Die Parameter der Frage werden wie folgt ausgelesen:

var mID = document.getElementById('mId').textContent;

Wenn der Code geändert wurde, muss mittels dem Command npx webpack der Code neu verpackt werden. Dabei entsteht der Ordner dist. Dieser Ordner ist im Root-Verzeichnis (code/) des Plugins abzulegen. Der alte Ordner ist vorher zu löschen. webpack ist bereits im node\_moduels Ordner vorhanden und muss nicht installier werden. Falls nicht vorhanden, muss npm installiert werden.

# Language Server

Die LS werden mittels Docker realisiert. Dabei hostet ein Docker einen Node-Server. Das Moodle-Plugin verbindet sich mit diesem Server. Der Server startet danach den jeweiligen LS. Die Kommunikation über das LSP erfolgt mittels eines Websocket mit Port 3001. Der Inhalt des Editors wird mittels eines HTTP-POST übertragen. Für die Implementierung eines neuen LS sind folgende Schritte notwendig:

1. Erweitern der index.js Datei um die entsprechende Dateiendung
2. Erstellen eines Docker-Containers welcher die Voraussetzungen des LS mitbringt
3. Anpassen der server.mjs Datei, um den LS zu starten.

Der Ordner node\_modules, welcher in allen vier Dockerordnern zu finden ist, wird vom Server benötigt und ist in die Docker-Images aufzunehmen. Da unter Linux-Dockern die, von Tools wie apt installierten, Node-Versionen veraltet sind, muss das Tool nvm verwendet werden. Eine mögliche Installation sieht so aus:

RUN curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.0/install.sh | bash

ENV NVM\_DIR="/root/.nvm"

ENV SH\_VERSION="1.38.0"

RUN . "$NVM\_DIR/nvm.sh" && nvm install 18.16.0

RUN . "$NVM\_DIR/nvm.sh" && nvm alias default stable

# Add NVM binaries to the system PATH

ENV PATH="/root/.nvm/versions/node/v18.16.0/bin:${PATH}"

Die bestehenden Docker sind wie folgt zu Starten:

docker run -d -p 3001:3001 -p 3002:3002 <name>

# Links

* <https://moodle.academy/>
* <https://moodledev.io/docs/apis/subsystems/form>
* <https://docs.moodle.org/dev/Question_types>
* <https://github.com/TypeFox/monaco-languageclient>
* <https://github.com/microsoft/monaco-editor>
* <https://microsoft.github.io/monaco-editor/>
* <https://microsoft.github.io/language-server-protocol/implementors/servers/>
* <https://www.npmjs.com/package/webpack>