

Aufgabe 1: MySQL Datenbank einrichten

Wir wollen das DBMS MySQL einsetzen, weil es weit verbreitet ist, kostenlos und viel Informationematerial im Internet darüber zu finden ist, wie man MySQL mit Node.js verbindet. Um einfach mit MySQL arbeiten zu können verwenden wir die lokale Server-Umgebung XAMPP.

a.) Lade XAMPP bei <https://www.apachefriends.org/de/index.html> herunter und installiere die Umgebung. Öffne anschließend das Control Panel und starte den Apache Webserver und den MySQL Datenbankserver. Öffne dann über den "Admin"-Button die Benutzerschnittstelle "phpMyAdmin".

b.) Lege eine neue Datenbank (DB) mit dem Namen "quizfight" an, die den Zeichensatz "utf16_bin" verwendet. Die DB benötigt zunächst eine Tabelle "questions" mit der Struktur aus Abbildung 1.

#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra
1	qid	int(11)			Nein	kein(e)		AUTO_INCREMENT
2	text	text	utf16_bin		Nein	kein(e)		
3	answer0	text	utf16_bin		Nein	kein(e)		
4	answer1	text	utf16_bin		Nein	kein(e)		
5	answer2	text	utf16_bin		Ja	NULL		
6	answer3	text	utf16_bin		Ja	NULL		
7	rightanswer	int(11)			Nein	kein(e)		

Abbildung 1: Tabelle "questions" der Datenbank "quizfight"

c.) Lege in der Tabelle "questions" mindestens 3 Fragen an.

Aufgabe 2: Datenbank Zugriff

Um mit unserem Server auf die DB zugreifen zu können brauchen wir einen MySQL-Treiber für Node.js und wir legen eine extra Klasse an, über die wir die Verbindung zur DB aufbauen. Wichtige Infos dazu findest du hier: <https://www.mysqltutorial.org/mysql-nodejs/connect/>

a.) Installiere den MySQL Treiber für Node.js.

b.) Definiere eine neue Klasse "DataController". Sie benötigt das Attribut "connection", das die Verbindung zur Datenbank enthält. Folge der Anleitung (s.o.) und baue eine Verbindung zur DB aus Aufgabe 1 auf.

c.) Der Server bindet die Klasse "DataController" ein und deklariert eine Variable "datacontroller". Innerhalb der "showReady"-Funktion wird diese Variable mit einem neuen Objekt der Klasse "DataController" initialisiert.

d.) Teste ob die Verbindung zur DB aufgebaut werden kann und die Meldung richtig angezeigt wird.

e.) Notiere dir die wichtigsten Informationen zu den Methoden "**connect**" und "**query**" in der "mysql"-Bibliothek von Node.js.

connect	
query	

Aufgabe 3: Server erweitern für Quiz

a.) Der Server übergibt beim Aufruf des Konstruktors von "QuizDuel" zusätzlich den "datacontroller", damit der Konstruktor von "QuizDuel" Daten aus der DB verwenden kann. Der Konstruktor muss natürlich auch angepasst werden.

b.) Informiere dich im Internet z.B. bei <https://www.mediaevent.de/javascript/callback-function.html> über **callback-Funktionen**, **anonyme Funktionen** und die Methode `bind()` (https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Global_Objects/Function/bind) bei JavaScript und notiere dir die wichtigsten Informationen.

callback-Funktion	
anonyme-Funktion	
<code>bind()</code>	

c.) Die Klasse "DataController" benötigt eine neue Methode "getRandomQuestion", die als Parameter eine callback-Funktion "cbfunc" erhält. Zur Abfrage einer Zufälligen Frage wird zunächst eine String Variable "sql" deklariert und initialisiert. Sie enthält eine SQL-Abfrage, die alle Attribute aller Datensätze der Tabelle "questions" aus der Datenbank liest. Anschließend wird mit der Methode "query" des "connection" Attributs eine zweite Methode "createQuestionObject" aufgerufen, die als Parameter "err", "result", "fields" und den callback-Parameter "cbfunc" des Aufrufers erhält. Die Methode "createQuestionObject" wirft einen Fehler, wenn der Parameter "err" nicht leer ist.

quest:Question

question = {text: „...“, answers: [„...“, „...“, „...“, „...“]}
rightAnswer = 0

Dann wird in einer neuen Variablen "r" ein Zufallswert mit der Methode `Math.random()` generiert, mit der Länge des Arrays "result" -1 multipliziert und mit der Methode `Math.floor` zu einer ganzen Zahl gerundet. So erhält man einen Zufälligen Index des Arrays "result", in dem alle Fragen der abgefragten Tabelle "question" stecken. Aus dem Element an dieser zufälligen Position wird ein neues Objekt "question" (siehe Objektdiagramm) zusammengestellt und an die callback-Funktion "cbfunc" als parameter übergeben.

Zusatzaufgaben

a.) Schreibe eine Anwendung, die ähnlich wie der quizFight-client mit express und socket.io arbeitet, aber unter einer anderen Adresse aufgerufen werden kann. Die Anwendung „quizFight-admin“ soll zunächst ein Passwort abfragen. Bei richtiger Passworteingabe wird ein Formular angezeigt, mit dem der Benutzer Fragen und Antworten in die Datenbank schreiben kann. Außerdem muss er markieren können welche Antwort richtig ist.