**Stand: 29.12.2016**

**Verwendete URLs im Client und ihre Funktionsbeschreibung**

*von Daniel Dobras*

***Kommentar:*** *grün markierte Bereiche funktionieren beiderseitig, die noch schwarz markierten warten auf eine Implementierung im Serverteil, um die bereits im Client implementierten Funktionen testen und anpassen zu können*

**Für die Registrierung:**

Neuen Benutzer erstellen

URL: serverIP + „user/create“

Übergabe von einem User: -1, username, password, false

Erwarte response: isUserNameOk, gibt mir zurück, ob der User schon registriert ist

**Für die Login:**

Registrierten Benutzer in das System einloggen

URL: user/login

Übergabe von einem User: -1, username, password, false

//kann hier noch nicht zwischen Lehrer und Schüler unterscheiden, deswegen übergebe ich standardmäßig einfach ein false bei isTeacher

Als Antwort erwarte ich folgende Attribute: isUserNameOk, isPasswortOk, isTeacher

**Für das Quiz:**

Fragen entsprechend einer ausgewählten Kategorie und Anzahl der Fragen

URL: listQuizQuestions

Übergabe von zwei Strings: category und numberOfQuestions

(Mit dem Befehl kannst du den Wert von numberOfQuestions zu einer number machen:

var total = +this.numberOfQuestions;)

Als Antwort krieg ich ein Question[ ] mit MultipleChoiceQuestions und OpenQuestions mit genau der größe, die ich in numberOfQuestions gefordert habe und allen Attributen, die wir definiert haben.

Die Frage selbst sieht so aus:

export class GeneralQuestion {

constructor(public id: number,

public category : any,

public questionType : string,

public question: string,

public answerA: string,

public answerB: string,

public answerC: string,

public answerD: string,

public correctAnswer: string) {}

}

Hieraus kann ich dann auslesen und meine HTML entsprechend anpassen

**Für das Verwalten der Fragen:**

**Erstellung von Fragen:**

**Offene Frage erstellen:**

URL: openquestion/create

Übergabe einer offenen Frage: var newOpenQuestion = new OpenQuestion(-1, this.category, this. question, this.correctAnswer);

Ansonsten wie Beispiel vom Auftraggeber

**MCQ erstellen:**

URL: multiplechoicequestion/create

Übergabe einer MCQ: var newMCQ = new MultipleChoiceQuestion(-1, this.category, this.question, this.answerA, this.answerB, this.answerC, this answerD, this.correctAnswer);

* Gib mir bei createOQ und createMCQ einfach das zurück, was im Beispiel zurückgegeben wurde

**Auflisten der Fragen:**

**Auflisten der Offenen Fragen:**

Funktioniert ja schon

**Auflisten der MCQ:**

URL: listMultipleChoiceQuestions

Als Antwort erwarte ich ein MCQ[ ] mit folgenden Attributen:

(public id: number,

public category : any, //each question has to have a category

public question: string,

public answerA: string,

public answerB: string,

public answerC: string,

public answerD: string,

public correctAnswer: number)

**Ändern von Fragen:**

Wird clientseitig erst mal nicht implementiert

* /question/change