

Testdrehbuch

Team:

Mitglied 1: Johannes Koch, 11709264

Mitglied 2: Andreas Oberhuber, 11722752

Mitglied 3: Lorenz Smidt, 11722387

Mitglied 4: Johannes Spies, 11722416

Mitglied 5: Simon Triendl, 11709265

Proseminargruppe:

PS Gruppe 3 - Team 5

Datum: 25.06.2019

1 Testabdeckung

Overall: 56,7% (method); 42,6% (line)

configs	89,5%
logic (QuizRoom)	58%
model (Database)	100%
services (Database)	76,7%
socket	6,9%
socket.events	100%
ui.beans	8,3%
ui.controllers	27,3%
utils	48,6%

2 JUnit Tests

2.1 Services:

Test Name	Beschreibung
AWTImageServiceTest	
testResizeImagePortraitJPG	Test ob die Bildbearbeitung für eine vertikale Zielauflösung für JPGs funktioniert.
testResizeImagePortraitPNG	Test ob die Bildbearbeitung für eine vertikale Zielauflösung für PNGs funktioniert.
testResizeImageLandscapeJPG	Test ob die Bildbearbeitung für eine horizontale Zielauflösung für JPGs funktioniert.
testResizeImageLandscapePNG	Test ob die Bildbearbeitung für eine horizontale Zielauflösung für PNGs funktioniert.

testSize1	Test ob das resultierende Bild die richtigen Dimensionen hat.
testSize2	Test ob das resultierende Bild die richtigen Dimensionen hat.
testStoreAvatarWithScale100X205	Test ob das resultierende Bild die richtigen Dimensionen hat.
testStoreAvatarInvalidType	Test auf nicht unterstützte/nicht existierende Dateitypen.
testStoreAvatarInvalidWidth	Test auf unzulässige Bildbreite.
testStoreAvatarInvalidHeight	Test auf unzulässige Bildhöhe.
testStoreAvatarInvalidOutputPath	Testet ob bei einem falschen Ausgabepfad die richtige Exception geworfen wird.
testStoreAvatarInvalidInputPath	Testet mit invalider Eingabe.
testZero	Testet ob Bilder der Größe 0x0px erstellt werden.
testNegativeWidth	Testet ob Bilder mit negativer Breite erstellt werden.
testNegativeHeight	Testet ob Bilder mit negativer Höhe erstellt werden.
EqualsImplementationTest	
testUserEqualsContract	Testet ob die equals und hashCode implementierung von User korrekt sind.
testUserRoleEqualsContract	Testet ob die equals und hashCode implementierung für die UserRole korrekt sind.
FileSystemStorageTest	
testInit	Testet ob die nötigen Ordner erstellt wurden.
testStoreAvatar	Testet ob das speichern eines Bildes funktioniert.
testStoreAnswer	Testet ob das speichern eines Bildes funktioniert.

testDeleteAllAnswersOfSet	Testet ob das löschen Aller Antworten von einem FrageSet funktioniert.
testEmptyAvatar	Testet das Speichern mit einer leeren Datei.
testEmptyAnswer	Testet das Speichern mit einer leeren Datei.
testLoadAvatar	Testet ob die gespeicherte Datei erfolgreich zurückgegeben wird.
testLoadDefaultAvatar	Testet ob die gespeicherte Datei erfolgreich zurückgegeben wird.
testLoadDefaultAnswer	Testet ob die gespeicherte Datei erfolgreich zurückgegeben wird.
testLoadAnswer	Testet ob die gespeicherte Datei erfolgreich zurückgegeben wird.
testDeleteAvatar	Testet ob das Löschen gespeicherter Dateien funktioniert.
testDeleteAnswer	Testet ob das Löschen gespeicherter Dateien funktioniert.
ImageApiControllerTest	
badAvatarRequestShouldReturnDefaultTest	Testet ob der Zugriff mit falschem Pfad das Standardbild zurückgibt.
badAnswerRequestShouldReturnDefaultTest	Testet ob der Zugriff mit falschem Pfad das Standardbild zurückgibt.
badAvatarRequestTest	Testet ob der Zugriff mit falschen File Extensions zu einem 400 Bad Request führt.
badAnswerRequestTest	Testet ob der Zugriff mit falschen File Extensions zu einem 400 Bad Request führt.
storeAndRequestAvatarTest	Test ob ein zuvor gespeichertes Bild erfolgreich zurückgegeben wird.
storeAndRequestAnswerTest	Test ob ein zuvor gespeichertes Bild erfolgreich zurückgegeben wird.
ManagerServiceTest	

testLoadAllManagers	Testet ob die richtige Anzahl an Managern in der Datenbank ist.
testGetManagerById	Testet ob ein Manager korrekt per id geladen werden kann.
testGetManagerOfPlayer	Testet ob ein Manager von einem Player geladen werden kann.
testGetManagerOfQuestionSet	Testet ob ein Manager von einem QuestionSet geladen werden kann.
testUpdateManager	Testet ob Änderungen an einem Manager nach speichern korrekt übernommen werden.
testUpdateManagerNoUser	Testet ob das Speichern eines Managers ohne User unmöglich ist.
testUpdateManagerNoEmail	Testet ob das Speichern eines Managers ohne EMail unmöglich ist.
testCreateManager	Testet ob das Erstellen eines neuen Managers möglich ist.
testCreateManagerNoEmail	Testet ob das Erstellen eines neuen Managers ohne EMail unmöglich ist.
testCreateManagerNoPassword	Testet ob das Erstellen eines neuen Managers ohne Passwort unmöglich ist.
testCreateManagerAlreadyUser	Testet ob das Erstellen eines neuen Managers mit einem bereits erstellten verbundenen Users unmöglich ist (User wird bei Erstellung automatisch generiert).
testCreateManagerWrongFunction	Testet ob das Speichern eines neuen Managers durch die generische saveManager Funktion unmöglich ist.
testDeleteManager	Testet ob das Löschen eines Managers möglich ist.
MathGeneratorTest	
testGetEasyQuestions	Testet ob die Generierung von Einfachen Questions funktioniert.
testGetHardQuestions	Testet ob die Generierung von Schweren Questions funktioniert.

PlayerServiceTest	
testLoadAllPlayers	Testet ob die richtige Anzahl an Playern in der Datenbank ist.
testGetPlayerById	Testet ob das Laden eines Players anhand der id funktioniert.
testGetPlayerByUsername	Testet ob das Laden eines Players anhand des Usernames funktioniert.
testGetPlayersOfManager	Testet ob das Laden aller Spieler eines Managers funktioniert.
testGetLastPlayedWith1	Testet ob die Anzahl an letzten Mitspielern stimmt für user3.
testGetLastPlayedWith2	Testet ob die Anzahl an letzten Mitspielern stimmt für user4.
testGetLastPlayedWith3	Testet ob die Anzahl an letzten Mitspielern stimmt für user5.
testAddPlayedWith	Testet ob das hinzufügen eines Players zur Liste der letzten Mitspieler über die Funktion für usernames funktioniert.
testAddPlayedWith2	Testet ob das hinzufügen eines Players zur Liste der letzten Mitspieler über die Funktion für Player funktioniert.
testUpdatePlayer	Testet ob das Aktualisieren eines Players in der Datenbank funktioniert.
testUpdatePlayerNoCreator	Testet ob das Aktualisieren eines Players in der Datenbank ohne Creator unmöglich ist.
testUpdatePlayerNoUser	Testet ob das Aktualisieren eines Players in der Datenbank ohne User unmöglich ist.
testCreatePlayer	Testet ob das Erstellen eines neuen Players möglich ist.
testUnauthorizedCreatePlayer	Testet ob das Erstellen eines neuen Players ohne Berechtigungen unmöglich ist.
testCreatePlayerNoCreator	Testet ob das Erstellen eines neuen Players ohne Creator unmöglich ist.

testCreatePlayerNoUsername	Testet ob das Erstellen eines neuen Players ohne username unmöglich ist.
testCreatePlayerNoPassword	Testet ob das Erstellen eines neuen Players ohne Passwort unmöglich ist.
testDeletePlayer	Testet ob das Löschen eines Players funktioniert.
testAddTotalScore	Testet ob das Hinzufügen von Punkten zum total score funktioniert.
testSetHighScore	Testet ob das Setzen eines neuen Highscores funktioniert.
testAddCorrectAnswers	Testet ob das Addieren von richtigen Antworten funktioniert.
testAddTotalAnswers	Testet ob das Addieren von gesamten Antworten funktioniert.
testAddPlayTime	Testet ob das Addieren von Spielzeit funktioniert.
testAddQSetPlays	Testet ob das Addieren von absolvierten Spielen von QuestionSets funktioniert.
testRemoveQSetPlays	Testet ob das Entfernen eines QuestionSets aus den Spielerstatistiken funktioniert.
testGetAccuracy	Testet ob die Berechnung der Accuracy funktioniert.
testGetRank	Testet ob die Berechnung des Rangs funktioniert.
testAddLastGameScore	Testet ob das Hinzufügen von letzten Spielständen funktioniert.
testAddLastGameScoresOverflow	Testet ob das Hinzufügen von letzten Spielständen funktioniert, auch wenn dadurch der Älteste gelöscht werden muss (limitiert auf die 10 letzten).
QuestionServiceTest	
testSaveNewQuestion	Testet ob das Speichern einer neuen Question funktioniert.

testUpdateQuestion	Testet ob das Aktualisieren einer Question funktioniert.
testSaveQuestionQuestionNull	Testet ob das Speichern einer leeren Question unmöglich ist.
testSaveQuestionQuestionSetNull	Testet ob das Speichern einer Question ohne QuestionSet unmöglich ist.
testSaveQuestionQuestionTypeNull	Testet ob das Speichern einer Question ohne QuestionSet unmöglich ist.
testSaveQuestionQuestionStringNull	Testet ob das Speichern einer Question ohne Frage unmöglich ist.
testSaveQuestionQuestionStringTooLong	Testet ob das Speichern einer Question mit zu langer Frage unmöglich ist.
testSaveQuestionAnswerStringNull	Testet ob das Speichern einer Question ohne richtige Antwort unmöglich ist.
testSaveQuestionAnswerStringTooLong	Testet ob das Speichern einer Question mit zu langer Antwort unmöglich ist.
testSaveQuestionWrongAnswerString_1Null	Testet ob das Speichern einer Question ohne 1. falsche Antwort unmöglich ist.
testSaveQuestionWrongAnswerString_1TooLong	Testet ob das Speichern einer Question mit zu langer 1. falschen Antwort unmöglich ist.
testSaveQuestionWrongAnswerString_2TooLong	Testet ob das Speichern einer Question mit zu langer 2. falschen Antwort unmöglich ist.
testSaveQuestionWrongAnswerString_3TooLong	Testet ob das Speichern einer Question mit zu langer 3. falschen Antwort unmöglich ist.
testSaveQuestionWrongAnswerString_4TooLong	Testet ob das Speichern einer Question mit zu langer 4. falschen Antwort unmöglich ist.
testSaveQuestionWrongAnswerString_5TooLong	Testet ob das Speichern einer Question mit zu langer 5. falschen Antwort unmöglich ist.
testGetAllByAnswerContaining	Testet ob das Finden aller Questions mit ähnlichen richtigen Antworten funktioniert.

testGetAllByQuestionContaining	Testet ob das Finden aller Questions mit ähnlichen Fragen funktioniert.
testGetAllByType	Testet ob das Finden aller Fragen eines Typs funktioniert.
testDeleteQuestion	Testet ob das Löschen einer Question funktioniert.
QuestionSetServiceTest	
testGetQuestionSetsOfManager	Testet ob das Laden aller QuestionSets eines Managers funktioniert.
testGetQuestionById	Testet ob das Laden eines QuestionSets anhand der id funktioniert.
testSaveQuestionSetNullSet	Testet ob das Speichern eines leeren QuestionSets unmöglich ist.
testSaveQuestionSetNullAuthor	Testet ob das Speichern eines QuestionSets ohne Author (= Manager) unmöglich ist.
testSaveQuestionSetNullName	Testet ob das Speichern eines QuestionSets ohne Namen unmöglich ist.
testSaveQuestionSetTooLongName	Testet ob das Speichern eines QuestionSets mit zu langem Namen unmöglich ist.
testSaveQuestionSetDescriptionNull	Testet ob das Speichern eines QuestionSets mit leerer Beschreibung unmöglich ist.
testSaveQuestionSetDescriptionTooLong	Testet ob das Speichern eines QuestionSets mit zu langer Beschreibung unmöglich ist.
testSaveQuestionSetDifficultyNull	Testet ob das Speichern eines QuestionSets ohne Schwierigkeitsgrad unmöglich ist.
testSaveQuestionSetSaveSet	Testet ob das Speichern eines korrekten QuestionSets funktioniert.
testSaveQuestionSetOverwriteSet	Testet ob das Aktualisieren eines QuestionSets in der Datenbank funktioniert.

testDeleteQuestionSet	Testet ob das Löschen eines QuestionSets aus der Datenbank funktioniert.
testGetAllContaining	Testet ob das Finden aller QuestionSets durch einen Teil des Namens funktioniert.
testGetAllByDifficulty	Testet ob das Finden aller QuestionSets eines Schwierigkeitsgrades funktioniert.
testGetQuestionSetOfQuestion	Testet ob das Finden des QuestionSets einer Frage funktioniert.
testGetAllQuestionSets	Testet ob das Laden aller QuestionSets in der Datenbank funktioniert.
QuizRoomTest	
testCreateQuizRoomEasy	Testet ob das Erstellen eines QuizRooms mit einfachem Schwierigkeitsgrad funktioniert.
testCreateQuizRoomHard	Testet ob das Erstellen eines QuizRooms mit schwerem Schwierigkeitsgrad funktioniert.
testCreateQuizRoomMath	Testet ob das Erstellen eines QuizRooms im MathGod Modus funktioniert.
testQuizRoomAddPlayer	Testet ob das Hinzufügen eines Players zu einem QuizRoom funktioniert.
SessionInfoBeanTest	
testLoggedIn	Testet ob der angemeldete User korrekt zurückgegeben wird.
testNotLoggedIn	Testet ob die Rückgabewerte bei nicht angemeldeten Usern stimmen.
UploadApiControllerTest	
testNoUser	Test ob richtig auf nicht eingeloggte User reagiert wird.
PlayerStatsControllerTest	
testNoUser	Test ob richtig auf nicht eingeloggte User reagiert wird.

UploadServiceTest	
testGetUploadNullUser	Test ob auf die Uploads zugegriffen kann wenn kein User eingeloggt ist.
testClearUploadNullUser	Test ob die Uploads gelöscht werden können wenn kein User eingeloggt ist.
testGetUploadPlayer	Test ob auf die Uploads zugegriffen kann wenn ein Player eingeloggt ist.
testGetUploadManager	Test ob auf die Uploads zugegriffen kann wenn ein Manager eingeloggt ist.
testClearUploads	Test ob das Löschen korrekt funktioniert.
UserServiceTest	
testDatainitialization	Testet ob die Ausgangsdaten die richtigen User enthalten.
testExistsUser	Testet ob User korrekt als existierend erkannt werden.
testDeleteUser	Testet ob das Löschen eines Users funktioniert.
testUpdateUser	Testet ob das Aktualisieren eines Users in der Datenbank funktioniert.
testCreateUser	Testet ob das Erstellen eines neuen Users funktioniert.
testExceptionForEmptyUsername	Testet ob das Speichern eines Users mit leerem username unmöglich ist.
testUnauthenticatedLoadUsers	Testet ob das Laden aller User ohne die benötigten Berechtigungen unmöglich ist.
testUnauthorizedLoadUsers	Testet ob das Laden aller User ohne die benötigten Berechtigungen unmöglich ist.
testUnauthorizedLoadUser	Testet ob das Laden eines Users ohne die benötigten Berechtigungen unmöglich ist.
testAuthorizedLoadUser	Testet ob das Laden eines Users mit den benötigten Berechtigungen

	funktioniert.
testUnauthorizedDeleteUser	Testet ob das Löschen eines Users mit den den benötigten Berechtigungen funktioniert.
QRWebSocketTest	Testen ob alle Konstruktoren und Zugriffsfunktionen der Klassen sinngemäß funktionieren.
TestAnswerJSON	
TestAssignQuestionEvent	
TestChatMessageJSON	
TestChatMessageEvent	
TestChatHistoryEvent	
TestClientEvent	
TestJokerUseEvent	
TestPlayerJSON	
TestPlayerJoinEvent	
TestPlayerKickEvent	
TestPlayerLeaveEvent	
TestPlayerTimeoutEvent	
TestReadyUpEvent	
TestRemoveQuestionEvent	
TestRoomInfoEvent	
TestScoreChangeEvent	
TestTimerSyncEvent	
TestGenericSocketEvent	

2.2 Beans / Controller / Utils / Converters:

Test Name	Beschreibung
AnswerPictureBeanTest	
getAnswers	Testet ob der Pfad richtig zurückgegeben wird
AvatarBeanTest	
getAvatar	Testet ob der Avatar des Spieler korrekt geladen wird (String path, String username)
getAvatar1	Testet ob der Avatar des Spielers korrekt geladen wird (Player player)
PasswordBeanTest	
passwordBeanTest	Testet ob das Passwort richtig kodiert wird
ProfileBeanTest	
testSetPlayer	Testet teilweise ob der Spieler der Liste (recently played with) hinzugefügt wird
PlayerDetailControllerTest	
deletePlayer	Testet ob der Spieler aus der Datenbank gelöscht wurde
ScrollPaginatorTest	
testUpdateList	Testet ob das updaten der Liste funktioniert.
testInitNext	Testet ob das initialisieren der nächsten Elemente funktioniert.
testIsInitialized	Testet ob das erste Element am Anfang schon initialisiert ist und ob die Größe der ersten Teilliste passt.
ScrollPartTest	
testGetterAndSetter	Testet ob Getter und Setter Methoden korrekt funktionieren

StatUtilTest	
testRoundPercentage	Testet ob eine Dezimalzahl richtig in Prozent umgerechnet und gerundet wird
testRound	Testet ob richtig gerundet wird
testGamesPlayed	Testet ob die Anzahl der gespielten Spiele eines Spieler korrekt sind
testMsToString	Testet ob Millisekunden korrekt in Minuten und Stunden umgerechnet und ausgegeben werden
CapitalizeConverterTest	
testCapitalize	Testet ob der erste Buchstabe eines Strings groß geschrieben wird

3 Selenium Tests

Tests ohne Anmeldung:

	Starting page	Buttons pressed		Target page
Test 1	Homepage	[Get Started]		Login page
Test 2	Homepage	[Signup Button]		Login page
Test 3	Homepage	[Coffee Button]		Buymeacoffee.com
Test 4	Homepage	[Donate Button]		Worldwildlife charity
Login Test	Loginpage	Insert credentials	[Submit Button]	Home

Tests als Manager:

	Starting page	Buttons pressed		Target page
Open Question Sets	Homepage	[QS Button]		Question Set Page
Create Question Set	Question Set Page	Insert 2 questions	[submit]	Question Set Page
Create New Player	Player Overview	Open Add User Form	Add User Credentials	Player Overview

Tests als Player:

	Starting page	Buttons pressed	Target page
Create Game	Homepage	[Create Game Button]	Game Lobby
Player Overview	Homepage	[Player Overview Button]	Player Overview