

de.htwberlin.kba.gr7.vocabduel.game_administration.export

Interface **GameService**

public interface **GameService**

Service for managing current games and their rounds.

Version:
1.0, May 2021
Author:
Sebastian Kehl, Lucas Larisch

Field Summary

Fields	
Modifier and Type	Field and Description
static int	NR_OF_ROUNDS Integer: Fix number of Rounds per game

Method Summary

All Methods Instance Methods Abstract Methods	
Modifier and Type	Method and Description
de.htwberlin.kba.gr7.vocabduel.game_administration.export.model.CorrectAnswerResult	answerQuestion (de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.User player, long gameId, int roundNr, int answerNr) Checks and stores the result for an answer submitted in a given round by a given player.
java.util.List<de.htwberlin.kba.gr7.vocabduel.game_administration.export.model.RunningVocabduelGame>	getPersonalChallengedGames (de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.User user) Collects all pending, i.e. not finished, games a given user has been challenged to and returns them.
int	removeWidowGames () Removes all running games with at least one removed user and all finished games with two removed users.
de.htwberlin.kba.gr7.vocabduel.game_administration.export.model.RunningVocabduelGame	startGame (de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.User playerA, de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.User playerB, java.util.List<de.htwberlin.kba.gr7.vocabduel.vocabulary_administration.export.model.VocableList> vocableLists) Starts a new game as playerA against playerB with the given parameters (languages/vocable lists) and returns an instance of that newly started match.
de.htwberlin.kba.gr7.vocabduel.game_administration.export.model.VocabduelRound	startRound (de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.User player, long gameId) Starts the next round of a given game as a given player, i.e. returns the respective next round object.

Field Detail

NR_OF_ROUNDS
static final int NR_OF_ROUNDS Integer: Fix number of Rounds per game

See Also:
Constant Field Values

Method Detail

startGame

```
de.htwberlin.kba.gr7.vocabduel.game_administration.export.model.RunningVocabduelGame startGame(de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.User playerA,
                                                                                               de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.User playerB,
                                                                                               java.util.List<de.htwberlin.kba.gr7.vocabduel.vocabulary_administration.export.model.VocableList> vocableLists)
throws de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.InvalidUserException,
       de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.InvalidGameSetupException,
       de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.NotEnoughVocabularyException,
       de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.UserOptimisticLockException,
       de.htwberlin.kba.gr7.vocabduel.vocabulary_administration.export.exceptions.VocabularyOptimisticLockException,
       de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.GameOptimisticLockException
```

Starts a new game as playerA against playerB with the given parameters (languages/vocable lists) and returns an instance of that newly started match.

Parameters:
playerA - Player initiating the game.
playerB - Opponent to be invited to play with playerA.
vocableLists - List of vocable lists to be used in the game. The questions will be randomly picked from these lists.

Returns:
New RunningVocabduelGame instance based on the given params.

Throws:
de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.InvalidUserException - One of the given users could not be found.
de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.InvalidGameSetupException - The setup of the game is not valid for a reason described in the given error message.
de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.NotEnoughVocabularyException - The provided VocableLists do not contain enough Vocables for GameAdministration.NR_OF_ROUNDS rounds per game.
de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.UserOptimisticLockException - An OptimisticLock error in the user module occurred.
de.htwberlin.kba.gr7.vocabduel.vocabulary_administration.export.exceptions.VocabularyOptimisticLockException - An OptimisticLock error in the vocabulary module occurred.
de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.GameOptimisticLockException - An OptimisticLock error in the game module occurred.

getPersonalChallengedGames

```
java.util.List<de.htwberlin.kba.gr7.vocabduel.game_administration.export.model.RunningVocabduelGame> getPersonalChallengedGames(de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.User user)
                                                                                               throws de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.GameOptimisticLockException
```

Collects all pending, i.e. not finished, games a given user has been challenged to and returns them.

Parameters:
user - User the pending/challenged matches of are to be returned.

Returns:
List of all unfinished games the given user has been challenged to or has started.

Throws:
de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.GameOptimisticLockException - An OptimisticLock error in the game module occurred.

startRound

```
de.htwberlin.kba.gr7.vocabduel.game_administration.export.model.VocabduelRound startRound(de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.User player,
                                                                                          long gameId)
```

throws de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.NoAccessException,
de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.GameOptimisticLockException

Starts the next round of a given game as a given player, i.e. returns the respective next round object.

Parameters:

player - Player the next round is to be returned for.

gameId - Id of the game the next round of is to be returned for the given user.

Returns:

Current VocabduelRound of a given game for a given user. including the 1 correct and the other wrong answer possibilities without knowing which is what.

Throws:

de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.NoAccessException - The given user is no participant of the given round or it could not be found at all.

de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.GameOptimisticLockException - An OptimisticLock error in the game module occurred.

answerQuestion

```
de.htwberlin.kba.gr7.vocabduel.game_administration.export.model.CorrectAnswerResult answerQuestion(de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.User player,
                                                    long gameId,
                                                    int roundNr,
                                                    int answerNr)
throws de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.InvalidVocabduelGameNrException,
de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.NoAccessException,
de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.GameOptimisticLockException
```

Checks and stores the result for an answer submitted in a given round by a given player. The information, which answer the right one is, stays server sided. This request checks whether the submitted answer is right or not and returns feedback for that answer incl. the correct answer in case of having submitted a wrong one. Afterwards only the check whether the user was wrong of right will be stored. The information, which answer the user chose, will be lost.

Parameters:

player - Player who has answered the question.

gameId - Id of the game the answer is to be submitted for.

roundNr - Nr of the round the answer is to be submitted for.

answerNr - Nr of the answer submitted by the given player.

Returns:

Result for the given round from the perspective of the given player incl. the correct answer in case of having submitted a wrong one.

Throws:

de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.InvalidVocabduelGameNrException - The question has already been answered by the current user or an invalid answer number has been stated.

de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.NoAccessException - The given user is no participant of the given round or it could not be found at all.

de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.GameOptimisticLockException - An OptimisticLock error in the game module occurred.

removeWidowGames

```
int removeWidowGames()
throws de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.GameOptimisticLockException
```

Removes all running games with at least one removed user and all finished games with two removed users.

Returns:

0 in case of success

Throws:

de.htwberlin.kba.gr7.vocabduel.game_administration.export.exceptions.GameOptimisticLockException - An internal error in the game module occurred.

