

# Anwendungsfälle

NvG, FH, SR

## 1 Einsatz eintragen

Die methode *insertPlayerToMatch(Player player, Match match)*; ist in *MatchService* implementiert. Falls alle Voraussetzungen erfüllt sind wird ein *LineUpEvent* generiert und dem Spiel hinzugefügt. Danach wird das Match gespeichert um die Änderungen konsistent zu machen. Nötige Voraussetzungen für das erstellen eines *LineUpEvents* sind:

- Der Spieler gehört zu einem der spielenden Teams.
- Das Team für das der Spieler spielt hat noch keine 11 Spieler in der Startaufstellung.

Um Spieler in das Spiel einzügen, nachdem das Spiel begonnen hat existiert die *substitutePlayers(Player playerOut, PlayerIn, Integer Minute, Match match)*; Hierbei wird der Spieler *PlayerOut* durch *PlayerIn* ersetzt, falls *PlayerOut* zur gegebenen *Minute* im Spiel(*match*) ist, beide in der selben Mannschaft spielen, die Mannschaft am Spiel teilnimmt und *PlayerIn* davor noch nicht gespielt hat.

Zum Austauschen der Spieler wird ein *SubstitutionEvent* erstellt und dem *Match* hinzugefügt.

## 2 Tor eintragen

Die Methode *insertGoal(GoalEvent goal, Player player, Match match)*; ist in der *MatchService*-Klasse implementiert. Beim *goal* müssen das Team, dass das Tor gutgeschrieben bekommt und die Spielminute eingetragen sein. Falls der Spieler im Match spielt, wird das Tor hinzugefügt. Ein Eigentor wird dadurch erstellt, dass im *GoalEvent* die Mannschaft des Gegners angegeben wird.

## 3 Ergebnis ausgeben

Die Methode *getResult(Match match)*; ist auch in *MatchService* implementiert. Es wird entweder das Zwischen- oder das Endergebnis ausgegeben, je nachdem, wann die Methode aufgerufen wird.

## 4 Gruppentabelle berechnen

Die Methode *printStandings(TournamentGroup group)*; ist in *GroupService* implementiert. Falls die Spiele in der Gruppe schon initialisiert wurden wird die Tabelle ausgegeben. Die Tabelle besteht 4 Zeilen , die aus Rang, Teamname, gespielten Spielen, Torverhältnis und Punkten bestehen.

## 5 Sieger ausgeben

Die Methode *weAreTheChampions(Tournament tournament)*; ist in *TournamentService* implementiert. Es werden die beiden Finalisten und der Dritte des Turniers ausgegeben.

## 6 aufrufende Funktionen

Alle Funktionen werden in den Testklassen ihres Service-Layers getestet. Die Testklassen befinden sich im Packet *dbs.project.service* im Ordner *src/test/* und heißen *{ KlasseServiceTest }* .java, wobei Klasse natürlich für die jeweilige Klasse steht.