Classes

Player

Fields: String name, int points, double multiplier, int challengeBalance, List<Run> runs(OneToMany)

Model for game DB table Player. Constructors with and without parameters. Methods fixMultiplier(), updateMultiplier() and updateCurrentPoints() for manipulation of private fields. Getters and setters working even if not written. ChallengeBalance is used to save the amount of point gains/losses through challenging others and being challenged.

PlayerRepository

Interface between DB and and rest of the application. Offers basic query functionality even without anything written in it. Autowired instances inside PlayerController and RunController.

PlayerController

Field: PlayerRepository pr

Responsible for player related functions. Adding/deleting players, showing personal pages and listing all players, reseting player points and multipliers and handling challenges between players. Deals with all request that have /player or /challenge present in URL. Additional methods challengePointSwap and resetPlayer.

Game

Fields: String name, int parTimeInSec, String description, List<Run> runs(OneToMany)

Model for game DB table Game. Has getters, setters and constructors with and without parameters. Additional method countPointTiers returns times for getting 5-0 points with given multiplier.

GameRepository

Interface between game table and rest of the application. Offers basic query functionality even without anything written in it. Autowired instances inside GameController and RunController.

GameController

Handles requests and data fetching related to games. Adding/Deleting, showing and listing of games. Requests to “/games/*”.

Run

Fields: int completionTimeInSec, int points, boolean current, Player player(ManyToOne), Game game(ManyToOne)

Model for game DB table Run. Has getters, setters and constructors with and without parameters. No additional methods. Variable current indicates if the run is related to the current event. Only current runs are noted when counting player points and multipliers.

RunRepository

Interface between run table and rest of the application. Offers basic query functionality even without anything written in it. Autowired instance inside RunController.

RunController

Fields: PlayerRepository pr, GameRepository gr, RunRepository rr

Requests and functionality relating to runs. Listing, deleting and adding runs. Also handles runs relevancy changing. All requests to “/runs/*”. Additional method countPointsOfRun.

DefaultController

Fields: PlayerRepository pr, GameRepository gr

Requests to root, data fetching for the form on the index page.

SBSRcounter

Main class for the whole application.