Testverbesserung- Dokumentation

Aufgabe 1

- races.csv einlesen

Fehler: es wurde zwar eine Tabelle erstellt aber nichts in die Tabelle eingetragen

Alter Code:

```
try {
            BufferedReader br = new BufferedReader(new
InputStreamReader(getClass().getResourceAsStream("/races.csv")));
            br.readLine();
            String line;
            while ((line = br.readLine()) != null) {
                String[] row = line.split(";");
                List<Race> races = this.em
                        .createNamedQuery("Race.getAll", Race.class)
                        .getResultList();
                Race currentRace;
                if (races.size()!=1){
                    currentRace = new Race();
                    this.em.persist(currentRace);
                }else{
                    currentRace = races.get(0);
                }

            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
```

Neuer Code:

```
BufferedReader br = new BufferedReader(new
InputStreamReader(getClass().getResourceAsStream("/races.csv")));

        try {
            br.readLine();
            String line;
            while ((line = br.readLine()) != null) {
                String[] row = line.split(";");
                Race r = new Race(Long.parseLong(row[0]), row[1],
                        LocalDate.parse(row[2],
DateTimeFormatter.ofPattern("dd.MM.yyyy")));
                em.persist(r);
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
```

- Teams und Driver einlesen

Fehler: Es wurden zwar alle Fahrer eingefügt, jedoch kamen dazu noch 420 leere Zeilen. Die Teams wurden richtig in die Tabelle eingfügt. UTF-8 fehlt wegen Kimi Räikkönen.

Alter Code:

```
private void readTeamsAndDriversFromFile(String teamFileName) {
        try {
            BufferedReader br = new BufferedReader(new
InputStreamReader(getClass().getResourceAsStream("/teams.csv"),
StandardCharsets.UTF_8));
            br.readLine();
            String line;
            while ((line = br.readLine()) != null) {
                String[] row = line.split(";");
                List<Team> teams = this.em
                        .createNamedQuery("Team.getTeambyName", Team.class)
                        .setParameter("NAME", row[0])
                        .getResultList();
                Team currentTeam;
                if (teams.size() != 1) {
                    currentTeam = new Team(row[0]);
                    this.em.persist(currentTeam);
                } else {
                    currentTeam = teams.get(0);
                }

                this.em.persist(new Driver(row[1], currentTeam));
                this.em.persist(new Driver(row[2], currentTeam));

            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();


        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

Grundsätzlich war der Code richtig jedoch wurden durch den Code in der Methode readResultsFromEndpoint der Klasse ResultsRestClient eingefügt. Deshalb habe ich diesen Code jetzt entfernt.

Neuer Code:

```
BufferedReader br = new BufferedReader(new
InputStreamReader(getClass().getResourceAsStream("/teams.csv"),
StandardCharsets.UTF_8));
```

Aufgabe 2

Fehler: Es wurde nichts in die Tabelle eingetragen und in der Driver Tabelle 420 zusätzliche leere

Zeilen eingetragen.

Alter Code:

```
Response response = this.target.request(MediaType.APPLICATION_JSON).get();
        JsonArray payload = response.readEntity(JsonArray.class);
        JsonObject object = payload.getJsonObject(0);
        List<JsonObject> values = payload.getValuesAs(JsonObject.class);
        for (JsonObject value : values){
            em.persist(new Result());
            em.persist(new Driver());
            System.out.println(value);
        }
```

Neuer Code: Zu der Driver Klasse musste ich eine Named Query hinzufügen

```
@NamedQueries({
        @NamedQuery(name = "Driver.findByName",
                query = "select d from Driver d where d.name = :NAME" ),
        @NamedQuery(
                name = "Driver.findAll",
                query = "select d from Driver d"
        )
})
```

readResultsFromEndpoint Methode:

```
public void readResultsFromEndpoint() {
        Response response = this.target.request(MediaType.APPLICATION_JSON).get();
        JsonArray payload = response.readEntity(JsonArray.class);
        persistResult(payload);
    }
```

persistResult Methode:

```
@Transactional
    void persistResult(JsonArray resultsJson) {

        for (JsonValue value : resultsJson){
            String name = value.asJsonObject().getString("driverFullName");
            int position = value.asJsonObject().getInt("position");
            Long raceNo = Long.parseLong("" + value.asJsonObject().getInt("raceNo"));

            em.persist(new Result(em.find(Race.class, raceNo),
                    position,
                    em.createNamedQuery("Driver.findByName", Driver.class)
                            .setParameter("NAME", name)
                            .getSingleResult()));
        }

    }
```

Aufgabe 3

Fehler: Ich bin nicht so weit gekommen.

Zuerst habe ich eine RestConfig Klasse erstellt

```
@ApplicationPath("api")
public class RestConfig extends Application {

}
```

Es muss ein Pfad im Result Endpoint eingefügt werden

```
@Path("results")
public class ResultsEndpoint {
.
.
.
}
```

Ein EntityManager muss erstellt werden

```
@PersistenceContext
EntityManager em;
```

Code für getPointsSumOfDriver Methode:

```
@GET
    @Produces(MediaType.APPLICATION_JSON)
    public JsonObject getPointsSumOfDriver(
            @QueryParam("name") String name
    ) {
        Long points = em
                .createNamedQuery("Result.sumPointsForDriver", Long.class)
                .setParameter("NAME", name)
                .getSingleResult();

        Driver driver = em
                .createNamedQuery("Driver.findByName", Driver.class)
                .setParameter("NAME", name)
                .getSingleResult();

        return Json
                .createObjectBuilder()
                .add("driver", driver.getName())
                .add("points", points)
                .build();
    }
```

Aufgabe 4

Fehler: Ich bin nicht so weit gekommen.

Man muss in der Driver Klasse eine NamedQuery erstellen um den Sieger des Rennens und das Land herauszufinden.

```
@NamedQuery(
        name = "Result.getWinnerOfRace",
        query = "select re.driver from Result re where re.position = 1 and " +
                "re.race = (select ra.id from Race ra where ra.country like :COUNTRY)"
)
```

findWinnerOfRace Methode:

```
 @GET
    @Produces(MediaType.APPLICATION_JSON)
    @Path("winner/{country}")
    public Response findWinnerOfRace(@PathParam("country") String country) {
        Long IdDriver = em.createNamedQuery("Result.getWinnerOfRace", Driver.class)
                .setParameter("COUNTRY", country)
                .getSingleResult()
                .getId();
        Driver winnerOfRace = em.find(Driver.class, IdDriver);
        return Response.ok(winnerOfRace).build();
    }
```

Aufagbe 5

Fehler: Ich bin nicht so weit gekommen.

Auch hier muss zuvor wieder eine NamedQuery erstellt werden. Diesmal in der Klasse Team.

```
@NamedQuery(
        name = "Result.racesWonByTeam",
        query = "select r.race from Result r where r.position = 1 " +
                "and r.driver in (select distinct d.id from Driver d " +
                "where d.team = (select distinct t.id from Team t where t.name =
:TEAM))"
)
```

Auch die Methode im ResultsEnpoint sieht ähnlich wie die vorherige aus.

Neuer Code:

```
@GET
    @Produces(MediaType.APPLICATION_JSON)
    @Path("raceswon")
    public List<Race> racesWonByTeam(@QueryParam("team") String team) {
        List<Race> wonRaces = em.createNamedQuery("Result.racesWonByTeam", Race.class)
                .setParameter("TEAM", team)
                .getResultList();
        return wonRaces;
    }
```

Aufgabe 6

Fehler: Nicht so weit gekommen.

Es muss eine NamedQuery in der Klasse Result implementiert werden.

```
@NamedQuery(
                name = "Result.getPoints",
                query = "select sum(r.points) from Result r where r.driver = :ID"
        )
```

Methode um List aller Fahrer mit ihren Punkten zu bekommen:

```
@GET
    @Produces(MediaType.APPLICATION_JSON)
    @Path("all")
    public List<String[]> allDriversWithPoints() {
        List<Driver> drivers = em
                .createNamedQuery("Driver.getDriver", Driver.class)
                .getResultList();
        List<String[]> driversWithPoints = new LinkedList<>();
        for (Driver driver : drivers) {
            Long points = em.createNamedQuery("Result.getPoints", Long.class)
                    .setParameter("ID", driver)
                    .getSingleResult();
            driversWithPoints.add(new String[]{driver.toString(), "" + points});
        }
        return driversWithPoints;
    }
```