

Verbesserung des 1. SEW-Tests

Aufgabe 1: Import CSV

readTeamsAndDriversFromFile

Ich habe die Daten direkt nach dem Splitten zu speichern, anstatt dass ich das String-Array an die persistTeamAndDrivers Methode weitergebe und dort die Daten erst speicher. Außerdem habe ich bei Driver versucht mit `map(a -> new Driver(a[1],new Team(a[0])))` die Daten zu speichern. Das funktioniert aber nicht, weil ich statt `new Team(a[0])` eine Array-Position statt einem Team hätte angeben müssen.

```
private void readTeamsAndDriversFromFile(String teamFileName) throws IOException {
    URL url = Thread.currentThread().getContextClassLoader()
        .getResource(teamFileName);
    try (Stream<String> stream = Files.lines(Paths.get(url.getPath()))) {
        stream
            .skip(1)
            .map(lines -> lines.split(";"))
            .map(a -> new Team(a[0]))
            .forEach(em::merge);

        //          stream
        //          .skip(1)
        //          .map(lines -> lines.split(";"))
        //          .map(a -> new Driver(a[1],new Team(a[0])))
        //          .forEach(em::merge);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Verbesserung:

```

private void readTeamsAndDriversFromFile(String teamFileName) throws IOException {
    URL url = Thread.currentThread().getContextClassLoader()
        .getResource(teamFileName);
    try (Stream<String> stream = Files.lines(Paths.get(url.getPath()),
        StandardCharsets.UTF_8)) {
        stream
            .skip(1)
            .map(lines -> lines.split(";"))
            .forEach(this::persistTeamAndDrivers);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

Hier wird zuerst mit der `@NamedQuery(name = "Team.findByName", query = "select t from Team t where t.name = :NAME")` geschaut, ob es das Team schon gibt, wenn nicht wird das Team gespeichert.

```

private void persistTeamAndDrivers(String[] line) {
    Team team = null;
    try {
        team = em.createNamedQuery("Team.findByName", Team.class)
            .setParameter("NAME", line[0])
            .getSingleResult();
    } catch (NoResultException ex) {
        team = new Team(line[0]);
        em.persist(team);
    }
    em.persist(new Driver(line[1], team));
    em.persist(new Driver(line[2], team));
}

```

Aufgabe 2: Import REST

readResultsFromEndpoint

```

public void readResultsFromEndpoint() {
    this.client = ClientBuilder.newClient();
    this.target = client.target(RESULTS_ENDPOINT);

    Response response = this.target.request(MediaType.TEXT_PLAIN).get();
    JSONArray payload = response.readEntity(JSONArray.class);

    persistResult(payload);
}

```

Verbesserung:

MediaType vom Response muss natürlich ein JSON sein, da payload ein JsonArray ist.

```
Response response = this.target.request(MediaType.APPLICATION_JSON).get();
```