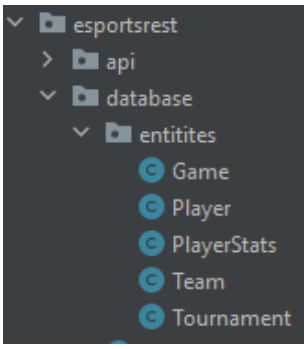
Übung 4-5 - Raci

Ich habe damit angefangen, ein Quarkus Projekt zu erstellen mithilfe der Quarkus webseite.

Als nächstes habe ich eine docker-compose file erstellt welche ine simple Postgresql Datenbank zum laufen bringt.

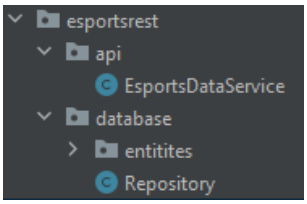
Weiters habe ich die Entities erstellt die ich für meine Applikation brauche.



Ein entity schaut beispielsweise so aus

```java
@Entity
@Table(name="game")
public class Game {
    @GeneratedValue
    @Id
    private long id;

    @OneToOne
    private Team firstTeam;

    @OneToOne
    private Team secondTeam;

    @OneToOne
    private Team winner;

    public Game() { }

    public Game(long id, Team firstTeam, Team secondTeam, Team winner) {
        this.id = id;
        this.firstTeam = firstTeam;
        this.secondTeam = secondTeam;
        this.winner = winner;
    }
```

Als nächstes habe ich den DataService und das repository für die Rest-requests erstellt, der Aufbau schaut folgendermaßen aus:

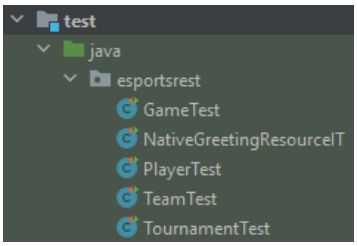


Im Repository habe ich dann Regions für alle Entities gemacht und dazu die grundlegensten Requests (GET, POST, DELETE). Diese werden vom DataService aufgerufen, der User schickt die Requests an den Service und

dieser schickt sie auch zurück.

```java
@ApplicationScoped
public class Repository {
    @Inject
    protected EntityManager em;

    Tournaments

    Games

    Players

    Teams
}
```

Zuletzt, zum testen des Backends, habe ich Tests zu jedem Entity geschrieben mit JUnit und REST-assured. Die Tests schauen grundsätzlich inhaltlich alle gleich aus. Die meisten Probleme beim testen lagen daran, dass ich meine Application natürlich so geschrieben habe, dass es keine doppelten Werte gibt. Nachdem ich aber die Datenbank selber erstellt habe schreibt er auch jedes mal drauf und löscht es nicht mehr also musste ich die Datensätze selbst löschen um create zu testen bis ich einen delete Test fertig hatte. Rückblickend wäre es wahrscheinlich schlauer gewesen zuerst die delete Tests zu schreiben.




```
All 11 tests are passing
```

Mein Projekt ist auf https://github.com/DionisRaci/Esports zu finden (Achtung: wenige commits).