

Formula 1

Version: 1.0

Kurzbeschreibung

Die Daten der Formel-1-Weltmeisterschaft 2018 liegen wie folgt vor:

- `aces.csv` ①
- `teams.csv` ①
- `results` ②

① `aces.csv` und `teams.csv` sind Dateien, die bereits im `resources`-Ordner im Projekt enthalten sind.

② Die Daten von `result` sind von einem REST-Endpoint abzurufen



Die Id's in der Datenbank-Tabelle `F1_RACE` werden nicht von der Datenbank generiert, sondern werden als Parameter mitgegeben.

Ihre Applikation soll diese Daten einlesen und Objekte der entsprechenden Klassen erzeugen. Mit deren Hilfe sollen anschließend folgende Auswertungen möglich sein:

- Gesamtpunkte eines Fahrers
- Fahrerweltmeister
- Konstrukteursweltmeister (=Teamweltmeister)
- Siegers eines bestimmten Rennens
- Liste der Rennen, die ein Fahrer / Team gewonnen hat

Aufgabenstellung

Aufgabe 1: Import CSV

Als erste Aufgabe sollen die CSV-Dateien `aces.csv` und `teams.csv` eingelesen werden. Hierfür sind bereits folgende Klassen implementiert:

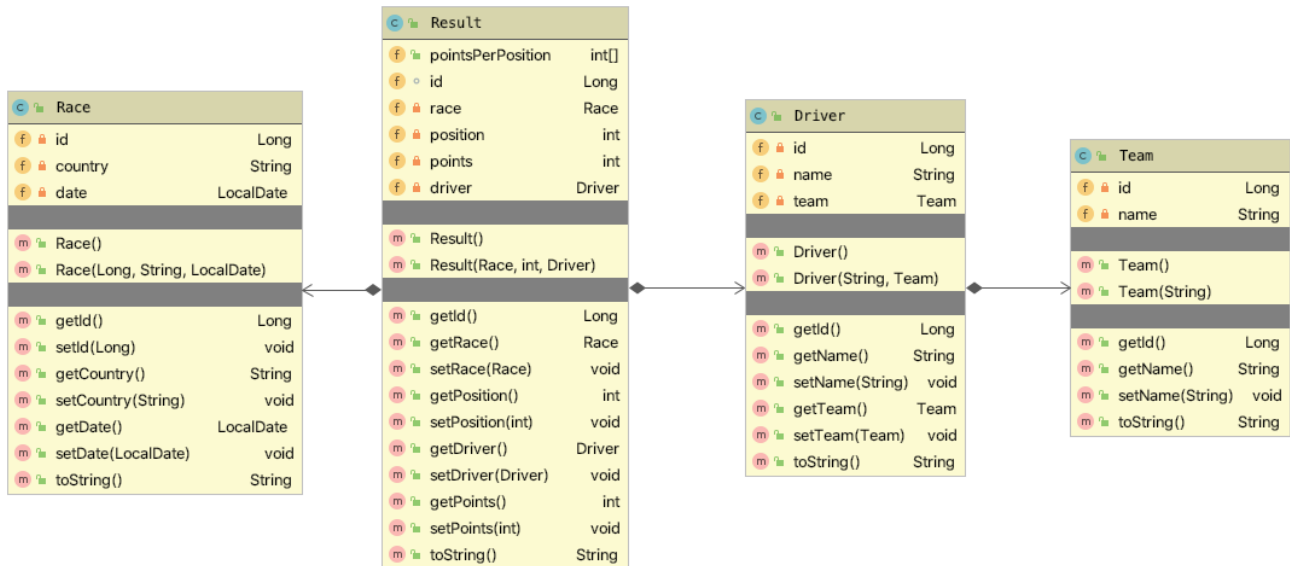
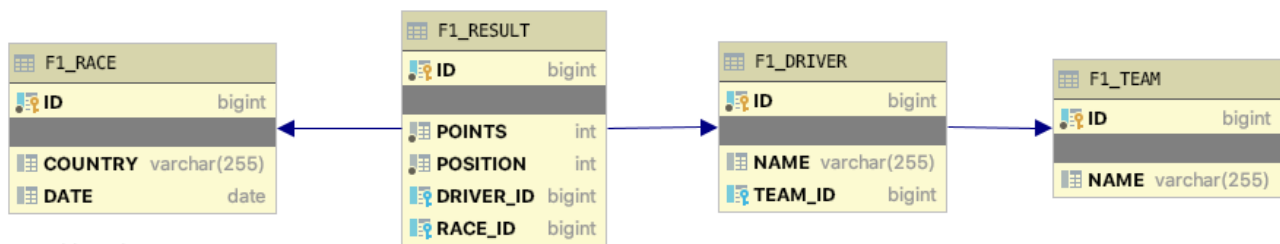


Figure 1. Klassendiagramm

Die Annotationen zum Speichern in der Datenbank sind noch hinzuzufügen.



Die Datei `teams.csv` wird auf zwei Klassen `Driver` und `Team` aufgeteilt.



Powered by yFiles

Figure 2. Entity-Relationship-Diagram (ERD)

Ergebnis

	ID	NAME
1	1	Ferrari
2	2	Force India
3	3	Haas
4	4	McLaren
5	5	Mercedes
6	6	Red Bull
7	7	Renault
8	8	Sauber
9	9	Toro Rosso
10	10	Williams

Figure 3. Tabelleninhalt F1_TEAM

	ID	NAME	TEAM_ID
1	1	Sebastian Vettel	1
2	2	Kimi Räikkönen	1
3	3	Sergio Perez	2
4	4	Esteban Ocon	2
5	5	Romain Grosjean	3
6	6	Kevin Magnussen	3
7	7	Fernando Alonso	4
8	8	Stoffel Vandoorne	4
9	9	Lewis Hamilton	5
10	10	Valtteri Bottas	5
11	11	Daniel Ricciardo	6
12	12	Max Verstappen	6
13	13	Nico Hulkenberg	7
14	14	Carlos Sainz	7
15	15	Charles Leclerc	8
16	16	Marcus Ericsson	8
17	17	Brendon Hartley	9
18	18	Pierre Gasly	9
19	19	Lance Stroll	10
20	20	Sergey Sirotkin	10

Figure 4. Tabelleninhalt F1_DRIVER

	ID	COUNTRY	DATE
1	1	Australia	2018-03-25
2	2	Bahrain	2018-04-08
3	3	China	2018-04-15
4	4	Azerbaijan	2018-04-29
5	5	Spain	2018-05-13
6	6	Monaco	2018-05-27
7	7	Canada	2018-06-10
8	8	France	2018-06-24
9	9	Austria	2018-07-01
10	10	Great Britain	2018-07-08
11	11	Germany	2018-07-22
12	12	Hungary	2018-07-29
13	13	Belgium	2018-08-26
14	14	Italy	2018-09-02
15	15	Singapore	2018-09-16
16	16	Russia	2018-09-30
17	17	Japan	2018-10-07
18	18	United States	2018-10-21
19	19	Mexico	2018-10-28
20	20	Brazil	2018-11-11
21	21	Abu Dhabi	2018-11-25

Figure 5. Tabelleninhalt F1_RACE

Aufgabe 2: Import REST

Als zweite Aufgabe sollen die Ergebnisse (Result) von einem gegebenen REST-Endpoint eingelesen werden. Implementieren Sie hierfür die Klasse `ResultRestClient`:

ResultsRestClient		
f	em	EntityManager
f	RESULTS_ENDPOINT	String
f	client	Client
f	target	WebTarget
m	readResultsFromEndpoint()	void
m	persistResult(JsonArray)	void

Link der Resource

<http://vm90.htl-leonding.ac.at/results>

Errechnen der Punkte

In der Tabelle sollen für jede Position die Anzahl der Punkte ermittelt werden. Diese werden bei einem Formel-1-Rennen folgendermaßen vergeben:

1. Platz = 25 Punkte
2. Platz = 18 Punkte
3. Platz = 15 Punkte
4. Platz = 12 Punkte
5. Platz = 10 Punkte
6. Platz = 8 Punkte
7. Platz = 6 Punkte
8. Platz = 4 Punkte
9. Platz = 2 Punkte
10. Platz = 1 Punkt

Ermitteln der Id des Fahrers

Für jeden Driver wird nach Namen in der Driver-Tabelle gesucht und zum Anlegen der Result-Objekte verwendet

Daten des Endpoints (teilweise)

JSON	Raw Data	Headers
Save	Copy	Collapse All
	Expand All	Filter JSON
▼ 0:		
driverFullName:	"Sebastian Vettel"	
position:	1	
raceNo:	1	
▼ 1:		
driverFullName:	"Lewis Hamilton"	
position:	2	
raceNo:	1	
▼ 2:		
driverFullName:	"Kimi Räikkönen"	
position:	3	
raceNo:	1	
▼ 3:		
driverFullName:	"Daniel Ricciardo"	
position:	4	
raceNo:	1	
▼ 4:		
driverFullName:	"Fernando Alonso"	
position:	5	
raceNo:	1	
▼ 5:		
driverFullName:	"Max Verstappen"	

Ergebnis

	ID	POINTS	POSITION	DRIVER_ID	RACE_ID
1	1	25	1	1	1
2	2	18	2	9	1
3	3	15	3	2	1
4	4	12	4	11	1
5	5	10	5	7	1
6	6	8	6	12	1
7	7	6	7	13	1
8	8	4	8	10	1
9	9	2	9	8	1
10	10	1	10	14	1
11	11	0	11	3	1
12	12	0	12	4	1
13	13	0	13	15	1
14	14	0	14	19	1
15	15	0	15	17	1
16	16	0	16	5	1
17	17	0	17	6	1
18	18	0	18	18	1
19	19	0	19	16	1
20	20	0	20	20	1
21	21	0	11	14	2
22	22	0	12	15	2
23	23	0	13	5	2
24	24	0	14	19	2
25	25	0	15	20	2
26	26	0	16	3	2
27	27	0	17	17	2
28	28	0	18	2	2
29	29	0	19	12	2
30	30	0	20	11	2
31	31	25	1	1	2
32	32	18	2	10	2
33	33	15	3	9	2
34	34	12	4	18	2
35	35	10	5	6	2

Figure 6. Tabelleninhalt (teilweise) F1_RESULT

Die Tabelle F1_RESULT enthält 420 Zeilen.

Aufgabe 3: Gesamtpunkte eines Fahrers

Als dritte Aufgabe sollen die Ergebnisse (Result) von einem gegebenen REST-Endpoint eingelesen werden. Implementieren Sie hierfür folgende Klasse mit der Methode `getPointsSumOfDriver(...)`:

Request (Beispiel)

```
GET http://localhost:8080/formula1/api/results?name=Lewis+Hamilton
```

Ergebnis



```
Run: results#1 x
GET http://localhost:8080/formula1/api/results?name=Lewis+Hamilton

HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: application/json
Content-Length: 40
Date: Thu, 14 Nov 2019 23:23:58 GMT

{
  "driver": "Lewis Hamilton",
  "points": 408
}

Response code: 200 (OK); Time: 619ms; Content length: 40 bytes
```

Kontrolle der Punkte

<https://www.motorsport-total.com/formel-1/ergebnisse/wm-stand/2018/fahrerwertung>

Aufgabe 4: : Siegers eines bestimmten Rennens

Request (Beispiel)

```
GET http://localhost:8080/formula1/api/results/winner/Spain
```

Aufgabe 5: Liste der Rennen, die ein Team gewonnen hat

Request (Beispiel)

```
GET http://localhost:8080/formula1/api/results/raceswon?team=Ferrari
```

Aufgabe 6 (für Spezialisten): Liste aller Fahrer mit ihren Punkten



Verwenden Sie als Rückgabebetyp bei mehreren Rückgabespalten der Query ein `List<Object[]>`

Request (Beispiel)

```
GET http://localhost:8080/formula1/api/results/all
```



Die Requests finden Sie in der Datei [/http-request/results.http](#)