

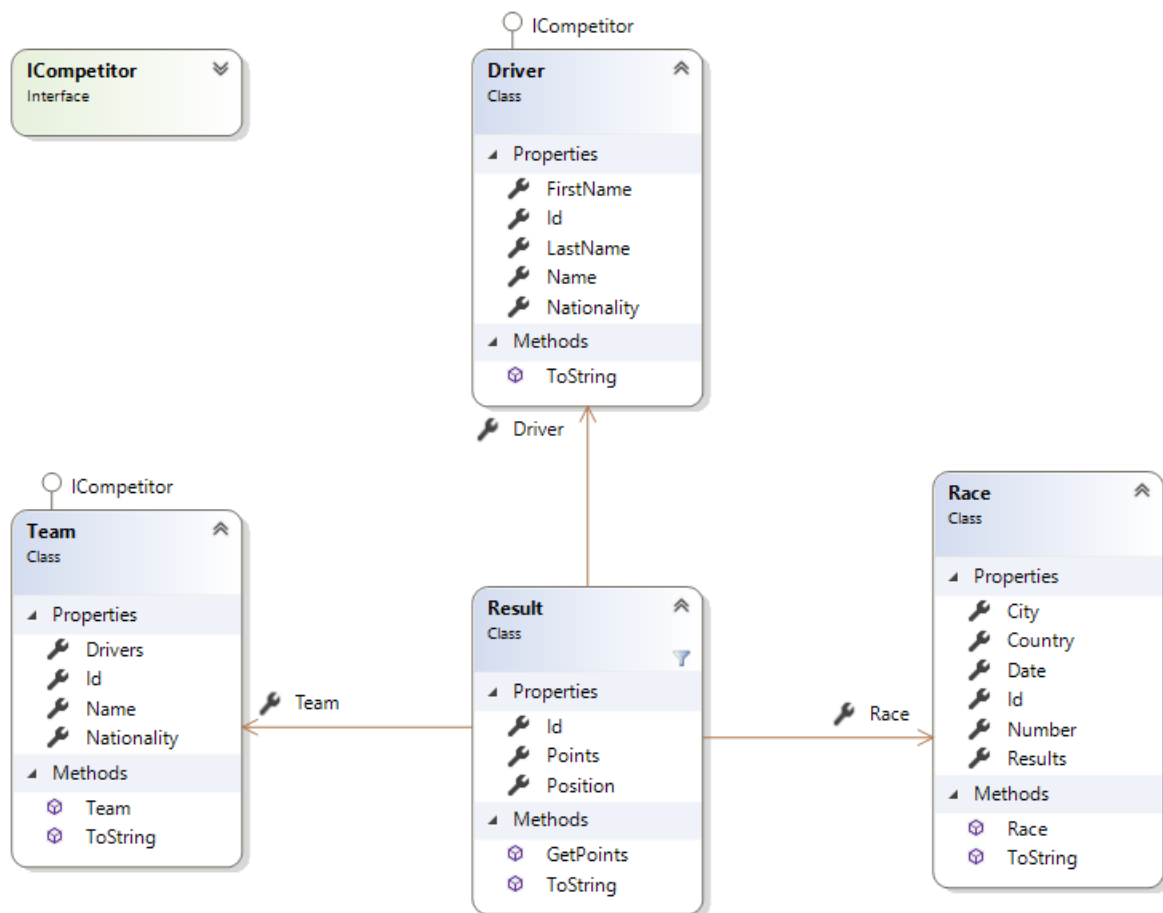
Entity Framework – Formula1 Importer

Lehrziele:

- Entity Framework
- Annotationen

Ziel dieser Übung ist die Grundlagen des Datenbankzugriffs per Entity Framework kennen zu lernen. Erweitern Sie dazu das Beispiel aus der Vorwoche um die Möglichkeit, die aus den XML-Dateien eingelesenen Informationen, in einer Datenbank zu persistieren.

Klassendiagramm



Stellen Sie mittels Annotationen folgende Einschränkungen sicher ([Hilfestellung](#)):

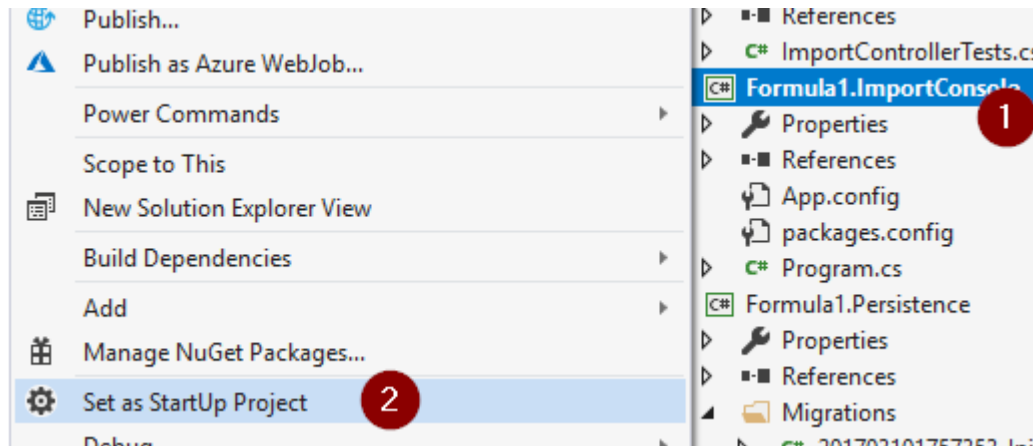
- Klasse *Driver*
 - *FirstName* (Pflichtfeld, max. Länge 50 Zeichen)
 - *LastName* (Pflichtfeld, max. Länge 50 Zeichen)
- Klasse *Race*
 - *Number* (Optional)

- *Country* (Pflichtfeld, max. Länge 200 Zeichen)
- *City* (Pflichtfeld, max. Länge 200 Zeichen)
- *Date* (Pflichtfeld, SQL Server Datentyp „date“)
- Klasse *Team*
 - *Name* (Pflichtfeld, max. Länge 50 Zeichen)
 - *Nationality* (Pflichtfeld, max. Länge 50 Zeichen)
- Klasse *Result*
 - *Race* (Pflichtfeld, Fremdschlüssel: RaceId)
 - *Team* (Pflichtfeld, Fremdschlüssel: TeamId)
 - *Driver* (Pflichtfeld, Fremdschlüssel: DriverId)

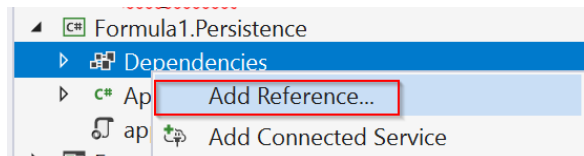
Die im Projekt „Formula1.PersistenceTests“ hinterlegten Unit Tests müssen erfolgreich ausgeführt werden können!

Erweiterten Sie die bestehenden Unit Tests mit eigenen Prüfungen.

1. Legen Sie das Projekt „Formula1.ImportConsole“ als Start-Projekt fest!



2. Fügen Sie die Abhängigkeiten zwischen den Projekten hinzu – denken Sie darüber nach, welches Projekt auf Elemente eines anderen Projektes zugreifen können muss (z.B. Formula1.Persistence muss auf Formula1.Core zugreifen können; da in der Persistence die einzelnen Entitäten „Driver“, „Team“, etc. verwendet werden).



3. Konfigurieren Sie die Verbindungseinstellungen in der Datei appsettings.json (analog zu „EF – First Demo“).

Inhalt:

```
{
  "ConnectionStrings": {
    "DefaultConnection": "Data Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=F1Importer;Integrated Security=True;"
  }
}
```

Hinweis:

Diese Datei muss in mehreren Projekten erstellt werden!
Build-Action beachten!

4. Fügen Sie mit dem Nu-Get Manager die notwendigen NuGet-Pakete hinzu (analog zu „EF – First Demo“)
5. Verwenden Sie die „Package Manager Console“ um mit folgenden Kommandos die Datenbank-Migrationen zu verwalten:
 - a. *Add-Migration <Name>*

b. Update-Database -Verbose

6. Verwenden Sie den „SQL Server Object Explorer“ um die erstellten Tabellen/Daten zu analysieren:

