

Formula 1 API (NodeJS - Examen) - 2025

🌟 Formule 1 RESTful API



🌟 Doel

Ontwikkel een **RESTful API** met **Node.js**, **Express**, **TypeScript** en **Mongoose** om gegevens over **Formule 1-races, teams, coureurs en circuits** te beheren. (Seizoen 2024)

📄 API Routes

🏎️ GET /races

- Haal alle races op.
- Breid de race-resultaten uit met details over de coureurs op basis van `driver_id`.
- Converteer de `countryCode` van een race naar een vlag-URL:

```
https://purecatamphetamine.github.io/country-flag-icons/3x2/COUNTRYCODE.svg
```

- Voeg deze vlag-URL toe aan het responsobject.
 - Tijdformaten:
 - Standaard worden tijden in milliseconden weergegeven.
 - Indien `?format=true` is meegegeven:
 - Tijd van positie **1** wordt omgezet naar `uren:minuten:seconden.milliseconden` (bijv. `1:31:44.742`). (3 cijfers na het punt)
 - Tijden van positie **2 en 3** worden omgezet naar seconden (bijv. `22.457`). (3 cijfers na het punt)
-

🌟 GET /teams

- Haal alle teams op.
- Breid de respons uit met coureur-details via `driver_id`.
- Converteer de `countryCode` van een team naar een vlag-URL:

```
https://purecatamphetamine.github.io/country-flag-icons/3x2/COUNTRYCODE.svg
```

- Voeg deze vlag-URL toe aan het responsobject.

GET /drivers

- Haal alle coureurs op.
- Ondersteunt **zoekfunctionaliteit** op `name` via een queryparameter:

```
?search=Verstappen
```

- **Case-insensitive** (bijv. `verstappen` en `Verstappen` geven hetzelfde resultaat).
- Converteer de `countryCode` van een coureur naar een vlag-URL:

```
https://purecatamphetamine.github.io/country-flag-icons/3x2/COUNTRYCODE.svg
```

- Voeg deze vlag-URL toe aan het responsobject.
-

GET /circuits

- Haal alle circuits op.
- Ondersteunt **zoekfunctionaliteit** op `name` via een queryparameter:

```
?search=Monaco
```

- **Case-insensitive** (bijv. `monaco` en `Monaco` geven hetzelfde resultaat).
-

Structuur

Werk met een gestructureerde opzet door gebruik te maken van **models, controllers en routes**:

- **Models**: Definieer de database-schema's met **Mongoose**.
 - **Controllers**: Beheer de logica van de API en verwerk aanvragen.
 - **Routes**: Bepaal de API-endpoints en verbind ze met de controllers.
-

Deployment

1. **Code publiceren**:
 - Push de code naar **GitHub**.
 - Voeg een **README** toe met uitleg over de API.
2. **Database opzetten**:
 - Maak een **MongoDB-database** aan (bijvoorbeeld via **MongoDB Atlas**).

3. API online zetten:

- Deploy de API via **Render**.
-

Indienen

- Dien de **GitHub-repository link** en de **Render live URL** in via **Syntra Cloud**.