F1 Grobal Tour API 및 백엔드 설계

- **사용자가 선택할 수 있는 세션 검색 시**

Year과 Countries의 선택값을 불러와 백엔드로 전달 백엔드는 아래 API를 사용해 사용할 수 있는 세션의 데이터 값을 전달

Sessions

Provides information about sessions. A session refers to a distinct period of track activity during a Grand Prix or testing weekend (practice, qualifying, sprint, race, ...).

HTTP Request

GET https://api.openf1.org/v1/sessions

Sample URL

https://api.openf1.org/v1/sessions?country_name=Belgium&year=2023

전달 API

```json
[
  {
    "meeting_key": 1216,
    "session_key": 9134,
    "location": "Spa-Francorchamps",
    "date_start": "2023-07-28T11:30:00+00:00",
    "date_end": "2023-07-28T12:30:00+00:00",
    "session_type": "Practice",
    "session_name": "Practice 1",
    "country_key": 16,
    "country_code": "BEL",
    "country_name": "Belgium",
    "circuit_key": 7,
    "circuit_short_name": "Spa-Francorchamps",
    "gmt_offset": "02:00:00",
    "year": 2023
  },
  {
    "meeting_key": 1216,
    "session_key": 9135,
    "location": "Spa-Francorchamps",
    "date_start": "2023-07-28T15:00:00+00:00",
    "date_end": "2023-07-28T16:00:00+00:00",
    "session_type": "Qualifying",
    "session_name": "Qualifying",
    "country_key": 16,
    "country_code": "BEL",
    "country_name": "Belgium",
    "circuit_key": 7,
```

```
    "circuit_short_name": "Spa-Francorchamps",
    "gmt_offset": "02:00:00",
    "year": 2023
  },
```


- **사용자가 특정 레이스 세션 선택 시**

session_key와 date_start와 date_end 값을 프런트 캐쉬에 저장

Drivers

Provides information about drivers for each session.

HTTP Request

GET https://api.openf1.org/v1/drivers

Sample URL

https://api.openf1.org/v1/drivers?&session_key=9158

통해 해당 세션 참가 드라이버 정보 전달


```json
[
  {
    "meeting_key": 1219,
    "session_key": 9158,
    "driver_number": 1,
    "full_name": "Max VERSTAPPEN",
    "name_acronym": "VER",
    "team_name": "Red Bull Racing",
    "team_colour": "3671C6",
  },
  {
    "meeting_key": 1219,
    "session_key": 9158,
    "driver_number": 2,
    "full_name": "Logan SARGEANT",
    "name_acronym": "SAR",
    "team_name": "Williams",
    "team_colour": "37BEDD",
  },
  {
    "meeting_key": 1219,
    "session_key": 9158,
    "driver_number": 4,
    "full_name": "Lando NORRIS",
```

```
    "name_acronym": "NOR",
    "team_name": "McLaren",
    "team_colour": "F58020",
},
```


1. 드라이버 위치 표시 API

Laps

Provides detailed information about individual laps.

HTTP Request

GET https://api.openf1.org/v1/laps

Sample URL

https://api.openf1.org/v1/laps?session_key=9161&lap_number=8

사용해서 전달


```json
[
  {
    "meeting_key": 1219,
    "session_key": 9161,
    "driver_number": 2,
    "lap_number": 8,
    "date_start": "2023-09-16T13:16:04.574000+00:00",
    "duration_sector_1": 27.263,
    "duration_sector_2": 39.523,
    "duration_sector_3": 26.466,
    "is_pit_out_lap": false,
    "lap_duration": 93.252,
  },
  {
    "meeting_key": 1219,
    "session_key": 9161,
    "driver_number": 77,
    "lap_number": 8,
    "date_start": "2023-09-16T13:16:11.261000+00:00",
    "duration_sector_1": 27.239,
    "duration_sector_2": 39.211,
    "duration_sector_3": 26.359,
    "i1_speed": 308,
    "i2_speed": 277,
    "is_pit_out_lap": false,
    "lap_duration": 92.809,
  },
  {
    "meeting_key": 1219,
```

```
    "session_key": 9161,
    "driver_number": 24,
    "lap_number": 8,
    "date_start": "2023-09-16T13:16:23.870000+00:00",
    "duration_sector_1": 27.319,
    "duration_sector_2": 39.339,
    "duration_sector_3": 26.6,
    "i1_speed": 309,
    "i2_speed": 277,
    "is_pit_out_lap": false,
    "lap_duration": 93.258,
  },
```


2. 드라이버 텔레메트리

Car data

Some data about each car, at a sample rate of about 3.7 Hz.

HTTP Request

GET https://api.openf1.org/v1/car_data

Sample URL

https://api.openf1.org/v1/car_data?driver_number=55&session_key=9159

로컬에 저장되어있는 session_key와 지금 현재 텔레메트리에 선택된 드라이버 넘버를 통해 드라이버 Car Data를 불러와 프런트로 데이터 전달


```json
[
  {
    "brake": 0,
    "date": "2023-09-15T13:08:19.923000+00:00",
    "driver_number": 55,
    "drs": 12,
    "meeting_key": 1219,
    "n_gear": 8,
    "session_key": 9159,
    "speed": 315,
    "throttle": 99
  },
  {
    "brake": 100,
    "date": "2023-09-15T13:35:41.808000+00:00",
    "driver_number": 55,
```

```
    "drs": 8,
    "meeting_key": 1219,
    "n_gear": 8,
    "session_key": 9159,
    "speed": 315,
    "throttle": 57
  }
]
```


Below is a table that correlates DRS values to its supposed interpretation (from FastF1).

DRS value	Interpretation
0	DRS off
1	DRS off
2	?
3	?
8	Detected, eligible once in activation zone
9	?
10	DRS on
12	DRS on
14	DRS on

3. 드라이버 인포 보드 랩타임, 인터벌등

인터벌은 4초마다 업데이트 따로 나머지 랩타임 등은 랩 완주 시 제공

Intervals

Fetches real-time interval data between drivers and their gap to the race leader. Available during races only, with updates approximately every 4 seconds.

HTTP Request

GET https://api.openf1.org/v1/intervals

Sample URL

https://api.openf1.org/v1/intervals?session_key=9165


```
[
  {
    "date": "2023-09-17T13:31:02.395000+00:00",
    "driver_number": 1,
    "gap_to_leader": 41.019,
```

```
    "interval": 0.003,
    "meeting_key": 1219,
    "session_key": 9165
  }
]
```


랩타임은 따로 랩 카운트 늘어날 -1의 값을 전달

Laps

Provides detailed information about individual laps.

HTTP Request

GET https://api.openf1.org/v1/laps

Sample URL

https://api.openf1.org/v1/laps?session_key=9161&lap_number=8


```*json*
[
  {
    "meeting_key": 1219,
    "session_key": 9161,
    "driver_number": 2,
    "lap_number": 8,
    "date_start": "2023-09-16T13:16:04.574000+00:00",
    "duration_sector_1": 27.263,
    "duration_sector_2": 39.523,
    "duration_sector_3": 26.466,
    "is_pit_out_lap": false,
    "lap_duration": 93.252,
    "segments_sector_1": [2048, 2048, 2049, 2048, 2048, 2048, 2049, 2048],
    "segments_sector_2": [2048, 2049, 2049, 2049, 2048, 2049, 2049, 2049],
    "segments_sector_3": [2049, 2048, 2051, 2048, 2048, 2049, 2048],
  },
  {
    "meeting_key": 1219,
    "session_key": 9161,
    "driver_number": 77,
    "lap_number": 8,
    "date_start": "2023-09-16T13:16:11.261000+00:00",
    "duration_sector_1": 27.239,
    "duration_sector_2": 39.211,
    "duration_sector_3": 26.359,
    "i1_speed": 308,
    "i2_speed": 277,
    "is_pit_out_lap": false,
    "lap_duration": 92.809,
```

```
    "segments_sector_1": [2048, 2048, 2049, 2049, 2048, 2048, 2049, 2049],
    "segments_sector_2": [2048, 2049, 2049, 2049, 2049, 2049, 2049, 2048],
    "segments_sector_3": [2049, 2049, 2048, 2049, 2049, 2048, 2048, 2048],
  },
  {
    "meeting_key": 1219,
    "session_key": 9161,
    "driver_number": 24,
    "lap_number": 8,
    "date_start": "2023-09-16T13:16:23.870000+00:00",
    "duration_sector_1": 27.319,
    "duration_sector_2": 39.339,
    "duration_sector_3": 26.6,
    "i1_speed": 309,
    "i2_speed": 277,
    "is_pit_out_lap": false,
    "lap_duration": 93.258,
    "segments_sector_1": [2049, 2049, 2049, 2049, 2048, 2049, 2048, 2049],
    "segments_sector_2": [2048, 2049, 2049, 2049, 2049, 2049, 2049, 2048],
    "segments_sector_3": [2048, 2048, 2048, 2049, 2048, 2049, 2049],
  },
```


Below is a table that correlates segment values to their meaning.

Value	Color
0	not available
2048	yellow sector
2049	green sector
2050	?
2051	purple sector
2052	?
2064	pitlane
2068	?

Segments are not available during races. Also, The segment values may not always align perfectly with the colors shown on TV, for unknown reasons.

- **메이저 포인트**

Lap_duration 값이 Null 인 경우 DNF로써 리타이어 처리 필요
Segments sector에서 2064 인 경우 in pit 처리 필요

타이어의 값은

Stints

Provides information about individual stints. A stint refers to a period of continuous driving by a driver during a session.

HTTP Request

GET https://api.openf1.org/v1/stints

Sample URL

https://api.openf1.org/v1/stints?session_key=9165

사용해 lap_start 값과 현재 리플레이 랩수 와 일치시 업데이트

```json
[
  {
    "meeting_key": 1219,
    "session_key": 9165,
    "stint_number": 1,
    "driver_number": 10,
    "lap_start": 1,
    "lap_end": 20,
    "compound": "MEDIUM",
    "tyre_age_at_start": 0
  },
  {
    "meeting_key": 1219,
    "session_key": 9165,
    "stint_number": 1,
    "driver_number": 81,
    "lap_start": 1,
    "lap_end": 20,
    "compound": "MEDIUM",
    "tyre_age_at_start": 0
  },
  {
    "meeting_key": 1219,
    "session_key": 9165,
    "stint_number": 1,
    "driver_number": 16,
    "lap_start": 1,
    "lap_end": 20,
    "compound": "SOFT",
    "tyre_age_at_start": 3
  },
```

4. 플래그 정보 전달

Race control

Provides information about race control (racing incidents, flags, safety car,

HTTP Request

GET https://api.openf1.org/v1/race_control

Sample URL

https://api.openf1.org/v1/race_control?date=2025-08-31

VSC 시작시


```
{
    "meeting_key": 1267,
    "session_key": 9920,
    "date": "2025-08-31T13:44:35+00:00",
    "driver_number": null,
    "lap_number": 31,
    "category": "SafetyCar",
    "flag": null,
    "scope": null,
    "sector": null,
    "message": "VIRTUAL SAFETY CAR DEPLOYED"
  },
```


VSC 엔딩시


```
  {
    "meeting_key": 1267,
    "session_key": 9920,
    "date": "2025-08-31T13:45:45+00:00",
    "driver_number": null,
    "lap_number": 32,
    "category": "SafetyCar",
    "flag": null,
    "scope": null,
    "sector": null,
    "message": "VIRTUAL SAFETY CAR ENDING"
  },
```


SC 시작시


```
{
    "meeting_key": 1267,
    "session_key": 9920,
    "date": "2025-08-31T14:12:19+00:00",
    "driver_number": null,
    "lap_number": 53,
    "category": "SafetyCar",
    "flag": null,
    "scope": null,
```

```
    "sector": null,
    "message": "SAFETY CAR DEPLOYED"
  },
  SC 엔딩시
  {
      "meeting_key": 1267,
      "session_key": 9920,
      "date": "2025-08-31T14:19:34+00:00",
      "driver_number": null,
      "lap_number": 57,
      "category": "SafetyCar",
      "flag": null,
      "scope": null,
      "sector": null,
      "message": "SAFETY CAR IN THIS LAP"
    },
  RED 플래그 발령 시
  {
      "meeting_key": 1143,
      "session_key": 7780,
      "date": "2023-03-31T02:25:43+00:00",
      "driver_number": null,
      "lap_number": null,
      "category": "Flag",
      "flag": "RED",
      "scope": "Track",
      "sector": null,
      "message": "RED FLAG"
    },
  RED 플래그 종료 시
  {
      "meeting_key": 1143,
      "session_key": 7780,
      "date": "2023-03-31T02:30:58+00:00",
      "driver_number": null,
      "lap_number": null,
      "category": "Flag",
      "flag": "CLEAR",
      "scope": "Track",
      "sector": null,
      "message": "TRACK CLEAR"
    },
```


여러 다른 레이스 컨트롤 메시지 있을 수 있지만 해당 메시지 json만 필터링해서 프론트로 전송

백엔드 서비스 처리 주요사항

1. 리얼타임 리플레이가 아니기에 사용자가 세션을 선택해서 리플레이를 재생하는 순간부터 필요한 정보를 API로 불러와 백엔드에서 캐싱처리
2. openF1의 API가 작동하지 않을 경우 에러 처리 메시지를 프론트로 전달해서 프론트는 해당 리플레이 기능이 작동할 수 없다는 에러메시지 출력후 리플레이 모드 나가기
3. 백엔드는 openF1의 API 응답 json을 그대로 가져와 필요한 정보만 남겨서 프론트로 전달하는 역할

백엔드 기술 스택 및 간단 설계서

좋아요! 😎 지금 상황을 정리하면:

- 기존 프로젝트: f1-global-tour replay 브랜치
- 새로운 레포지토리: **백엔드 전용**
- 요구사항: **실시간 API 호출 및 처리**

그러면 레포지토리 생성부터 초기 설정, 기술 스택까지 단계별로 정리해드릴게요.

1️⃣ 레포지토리 생성

1. GitHub에서 새 레포지토리 생성
    - 이름: f1-global-tour-backend
    - **Private/ Public** 선택
    - README.md 초기화 선택 가능
2. 로컬에 클론
3. git clone https://github.com/<username>/f1-global-tour-backend.git
4. cd f1-global-tour-backend

2️⃣ 기술 스택 추천

핵심 요구사항 기준

- **실시간 API 호출 및 처리** → 비동기/논블로킹 환경이 유리
- **백엔드 전용** → REST API 또는 GraphQL 제공
- 데이터 처리/저장 필요 가능

추천 스택

영역	추천
런타임	**Node.js 22+** (비동기 처리에 강점)
프레임워크	**NestJS** (TypeScript 기반, 모듈화 좋음) 또는 **Express.js** 단순 구현 가능
DB	**PostgreSQL** (관계형), **Redis** (실시간 캐싱/큐)
API 요청	**Axios** 또는 **node-fetch**
실시간 처리	**Socket.io** (웹소켓), **BullMQ** (큐 기반 처리)

영역	추천
코드 품질	**ESLint + Prettier**
테스트	**Jest**

NestJS를 쓰면 모듈 단위로 관리하기 좋고, TypeScript라 기존 프론트와 타입 공유 가능 (예: shared-types 패키지 활용)

---

3 초기 프로젝트 설정 (NestJS 기준)

```
# NestJS CLI 설치
npm install -g @nestjs/cli

# 새 프로젝트 생성
nest new f1-global-tour-backend
# Package manager: pnpm / npm 선택 가능
```

- 디렉토리 구조가 깔끔하게 잡힘
- 기본적인 REST API 서버 바로 시작 가능
- TypeScript 지원 기본

---

4 추가 설정

1. **환경 변수 관리**
    - .env 파일 생성
    - ex)
    - DATABASE_URL=postgres://user:pass@localhost:5432/f1db
    - API_KEY=xxxx
    - @nestjs/config 설치 후 사용
2. **공유 타입 적용**
    - 기존 프론트 @f1-tour/shared-types 패키지 npm/pnpm workspace로 가져오기
    - API 요청/응답 타입 통일 가능
3. **API 호출**
    - Axios 설치:
    - pnpm add axios
    - 서비스 단에서 비동기 호출:
    - const response = await axios.get('https://api.example.com/data');
4. **실시간 처리/큐**
    - 실시간 이벤트 처리: Socket.io
    - 주기적/대량 API 처리: BullMQ + Redis

---

5 Git & CI/CD 초기 설정

1. .gitignore
2. node_modules
3. dist
4. .env
5. GitHub Actions로 CI/CD 가능
    - Push 시 빌드, 테스트 자동 실행

- Dockerize 가능하면 서버 배포 편리

정리하면:

- **Node.js + NestJS + TypeScript** → 안정적, 모듈화, 타입 공유
- **PostgreSQL + Redis** → 실시간 처리 및 캐싱
- **Axios / Socket.io / BullMQ** → API 호출 + 실시간 이벤트 + 큐 처리
- **pnpm workspace 활용** → 프론트와 타입 공유