

開発者向け仕様書：F1 2025 システム

25G1058 佐藤 樹

2025 年 12 月 28 日

1 システム概要

本プログラムは、F1 2025 年シーズンのコンストラクター情報を管理するための Web アプリケーションです。Express によるサーバサイド処理と EJS テンプレートを組み合わせ、データの動的な表示と追加機能を実現しています。

2 データ構造

データはサーバ側の変数 `team`（オブジェクト配列）に格納されます。

- `id`: 順位を示す数値または文字列。
- `code`: ポイント数を格納する文字列。
- `name`: チーム名を示す文字列。
- `image`: 画像ファイルへのパス（`public/`配下）。

3 リソースと機能詳細

本システムは、以下の HTTP メソッドとリソース名を用いて操作を定義しています。

機能	HTTP メソッド	リソース名	処理内容
一覧表示	GET	/F1	全データの抽出と一覧表示
データ追加	GET	/F1.add	クエリパラメータを用いた新規登録
データ削除	DELETE (予定)	/F1/:id	指定 ID のデータ削除
データ編集	PUT (予定)	/F1/:id	指定 ID のデータ更新

4 ページ遷移と制御フロー

4.1 遷移の仕組みとユーザー導線

利用者が `http://localhost:8080/F1` にアクセスすると一覧画面が描画されます。データの追加は `/F1_add` を通じて行われます。チーム名をクリックするなどのアクションにより、詳細情報の取得や画面遷移が発生する構造となっています。

4.2 操作後の表示内容

追加・変更・削除の各操作が完了した後は、サーバ側で一覧画面へ強制的に遷移させます。これにより、利用者は常に最新のデータ状態を確認することが可能です。

5 授業で説明していない技術の概要と採用理由

本システムの開発にあたり、講義資料の範囲外で採用した技術について以下に記します。

1.REST API（設計思想）

- **概要:** URL を「リソース (物)」として扱い、HTTP メソッド (GET/-POST/PUT/DELETE) で操作を決定する設計手法です。
- **採用理由:** 「追加なら `_add`」「削除なら `_delete`」と個別に名前を付けるのではなく、統一されたルールで URL を設計することで、プログラムの拡張性と可読性を高めるため採用しました。

2.res.redirect()（リダイレクト機能）

- **概要:** 処理完了後、サーバからブラウザに対して別の URL へ再アクセスするように指示を出す機能です。
- **採用理由:** データ追加後に同じ画面で「更新」ボタンを押すと二重にデータが登録されるのを防ぎ、スムーズに一覧画面へ戻るユーザー体験 (UX) を提供するため採用しました。

3.express.static（静的ファイル配信設定）

- **概要:** サーバ上の特定のフォルダ（本システムでは `public`）内のファイルを、外部から直接参照可能にするミドルウェアです。
- **採用理由:** チームロゴや車両画像などの画像資産を、EJS テンプレートから効率的に呼び出し、安定して表示させるために不可欠な技術であるため採用しました。

4."use strict";（厳格モード）

- **概要:** JavaScript の記述ミスエラーとして検出しやすくする宣言です。

- **採用理由:** 意図しないグローバル変数の作成などを防ぎ、プログラムの動作を安定させ、バグの混入を未然に防ぐため採用しました。

6 ソースコードの管理

本システムのソースコードは、以下の GitHub リポジトリで管理しています。

https://github.com/satoitsuki1121/webpro_06.git