

一覧表示システム
Web アプリケーション仕様書

25G1034 恩田隼士

2025 年 12 月 25 日

目次

| | |
|--------------------------------|----------|
| 第1章 開発者向け仕様書：星座一覧表示システム | 2 |
| 1.1 概要 | 2 |
| 1.2 データ管理 | 2 |
| 1.2.1 データ構造 | 2 |
| 1.2.2 表示における制約事項 | 2 |
| 1.3 ディレクトリ構成 | 3 |
| 1.4 HTTP メソッドとルーティング | 3 |
| 1.5 ページ遷移 | 4 |
| 1.6 各機能・リソース詳細 | 5 |
| 1.6.1 トップページ | 5 |
| 1.6.2 一覧表示機能 | 5 |
| 1.6.3 新規作成機能 | 6 |
| 1.6.4 詳細表示機能 | 6 |
| 1.6.5 編集・更新機能 | 6 |
| 1.6.6 削除機能 | 7 |

ソースコード

本仕様書で使⽤したソースコードを添付する。

https://github.com/crab2424/webpro_submit

第1章 開発者向け仕様書： 星座一覧表示システム

1.1 概要

本仕様書は、Node.js およびテンプレートエンジン EJS を用いた「星座一覧表示システム」の設計仕様書である。本システムは、サーバーサイドで星座データを管理し、EJS を用いて動的に HTML を生成・表示する。データベースの利用は行わず、サーバープロセスのメモリ上（変数）でデータを保持・操作することを前提とする。

1.2 データ管理

1.2.1 データ構造

サーバー内の配列変数で管理するデータ構造は表 1.1 の通りに構成する。

| 表 1.1: 星座データ構造 | | |
|----------------|--------|--------|
| プロパティ名 | データ型 | 説明 |
| id | Number | 一意な識別子 |
| name | String | 星座名 |
| en | String | 英語表記 |
| shape | String | 星座の形 |
| height | String | 高度 (°) |
| star | String | 代表する星 |
| season | String | 季節 |

id を除くすべてのデータは入力フォームで作成、編集が可能である。id はデータ配列の長さに基づいて自動採番するため、作成時に入力不要である。

1.2.2 表示における制約事項

本システムでは、データ id と配列インデックスの整合性を保つため、以下の仕様を採用する。

- id とインデックスの一致: 配列のインデックスと ID の扱いを簡易化させるため、配列のインデックス 0 番目にはダミーデータを作成し、システム上では非表示とする。
- ID とインデックスの関係: データの削除操作により、配列のインデックスと各データの ID は必ずしも一致しないものとす。そのため、各機能におけるデータの特定は ID の値を基準に行う。
- 一覧表示の処理: id が 1 番のデータから表示させるため、EJS テンプレート側で配列操作 (data.slice(1)) を用いて表示する。

1.3 ディレクトリ構成

本システムは、図 1.1 に示すディレクトリ構造に従って、使用するファイルを配置する。

```
webpro_submit/
├─ app/
│   ├── app_system.js      (メインロジック・データ変数保持)
│   ├── public/           (静的ファイル)
│   │   └─ stella_new.html (新規作成フォーム)
│   └─ views/             (EJS テンプレート)
│       ├── stella/       (星座システム)
│       │   ├── stella_check.ejs (削除確認画面)
│       │   ├── stella_detail.ejs (詳細表示画面)
│       │   ├── stella_edit.ejs  (編集フォーム)
│       │   └─ stella.ejs      (一覧表示画面)
│       └─ landing.ejs      (トップページ)
```

図 1.1: ディレクトリ構成

1.4 HTTP メソッドとルーティング

本システムにおける各 URL と HTTP メソッド、および対応する処理を表 1.2 に定義する。

表 1.2: ルーティング一覧

| 機能 | メソッド | パス (URL) | 対応ビュー |
|----------|------|--------------------|-------------------------|
| トップページ | GET | / | views/landing.ejs |
| 一覧表示 | GET | /stella | views/stella/stella.ejs |
| 新規作成フォーム | GET | /stella/create | public/stella_new.html |
| 詳細表示 | GET | /stella/:id | views/stella_detail.ejs |
| 編集フォーム | GET | /stella/edit/:id | views/stella_edit.ejs |
| 削除確認 | GET | /stella/check/:id | views/stella_check.ejs |
| 新規データ作成 | POST | /stella | 処理後一覧を表示 |
| 新規データ作成 | POST | /stella/create | 処理後新規作成へリダイレクト |
| データ更新 | POST | /stella/update/:id | 処理後詳細ページを表示 |
| データ削除 | GET | /stella/delete/:id | 処理後一覧へリダイレクト |

1.5 ページ遷移

本システムにおける画面間の遷移を図 1.2 に示す。なお、本システムのページには戻るリンクを配置するため、一覧表示ページ及び詳細表示ページに直接遷移することが可能である。

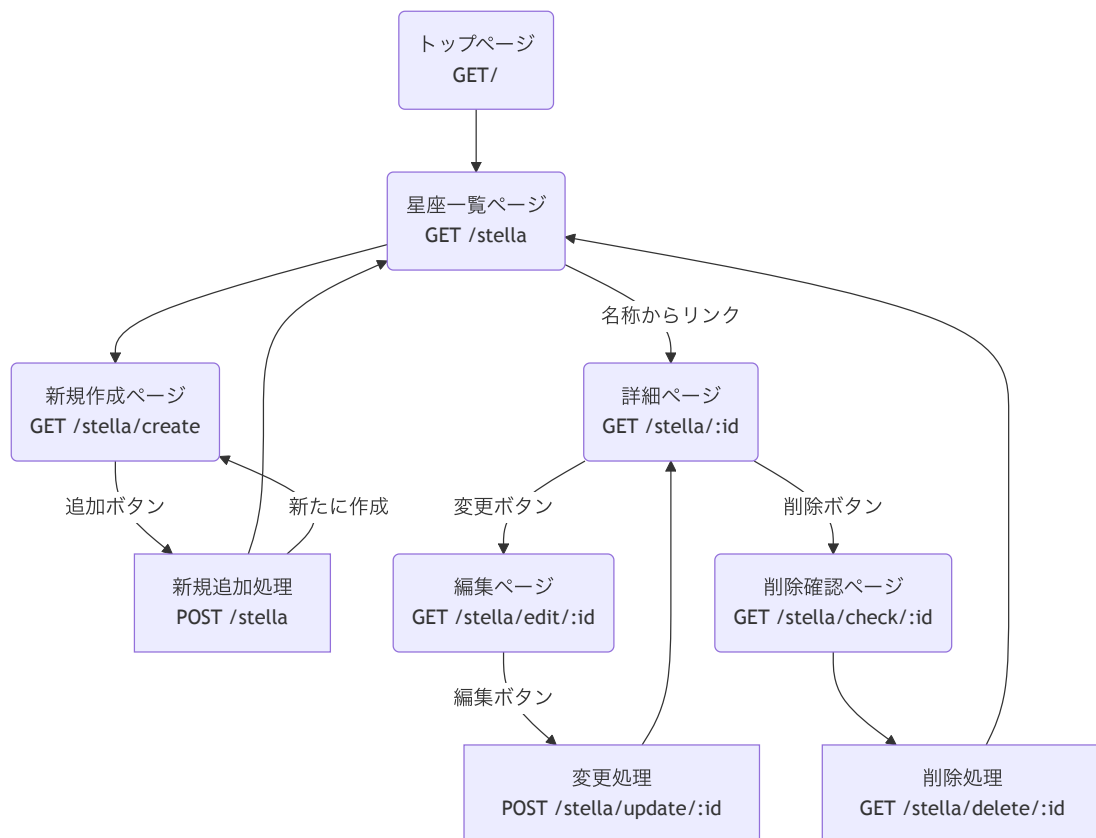


図 1.2: 画面遷移フローチャート

1.6 各機能・リソース詳細

1.6.1 トップページ

- URL: /
- 処理: views/landing.ejs を表示する.
- 要素: 各システムへのリンク

1.6.2 一覧表示機能

- URL: /stella
- 処理: サーバー変数の全データを EJS に渡し, for 文を用いてテーブル形式で表示する.
- 要素: 各行ごとの要素の名前に対応する詳細リンク, 追加ボタン

1.6.3 新規作成機能

- フォーム: GET /stella/create
 - 処理: 表 1.1 の id を除くすべてのプロパティを入力フィールドとして表示する.
 - 送信先: POST /stella
 - 要素: 各プロパティに対応する入力フィールド, 登録ボタン, 登録後新規作成ボタン
- 作成処理: POST /stella
 - リクエストボディから値を取得.
 - 新しい id を採番し, サーバー変数 (配列) に push する.
 - 新規作成したデータの内容をサーバーのターミナルにログ出力する.
 - 処理完了後, 一覧画面 (/stella) へリダイレクトする.

1.6.4 詳細表示機能

- URL: /stella/:id
- 処理: URL パラメータの id に基づき配列を検索し, 対象データを表示する.
- 要素: 編集ボタン, 削除ボタン, 一覧に戻るリンク

1.6.5 編集・更新機能

- フォーム: GET /stella/edit/:id
 - 処理: 対象データを検索し, value 属性に現在の値を埋め込んで表示する.
 - 送信先: POST /stella/update/:id
- 更新処理: POST /stella/update/:id
 - id に基づき配列内の該当インデックスを特定.
 - リクエストボディの値でプロパティを上書きする.
 - 表示: 更新内容をサーバーのターミナルにログ出力する.
 - 更新後, 詳細画面 (/stella/:id) を表示する.

1.6.6 削除機能

- フォーム: GET /stella/check/:id
 - 処理: 簡易確認フォームを表示する.
 - 送信先: GET /stella/delete/:id
- 削除処理: GET /stella/delete/:id
 - idに基づき配列から splice で要素を取り除く.
 - 削除した要素の名前をサーバーのターミナルにログ出力する.
 - 削除後, 一覧画面 (/stella) へリダイレクトする.