

# 都道府県一覧表示システム：開発者向け仕様書

25G1034 恩田 隼士

2025 年 12 月 19 日

## 1 概要

本ドキュメントは、Node.js およびテンプレートエンジン EJS を用いた「都道府県管理 Web アプリケーション」の設計仕様書である。本システムは、サーバーサイドで都道府県データを管理し、EJS を用いて動的に HTML を生成・返却する。データの永続化（DB 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。

## 2 システム要件

### 2.1 技術スタック

- ランタイム: Node.js
- Web フレームワーク: Express
- テンプレートエンジン: EJS
- データ保存先: サーバーサイドメモリ（配列変数）

### 2.2 データ構造 (Schema)

サーバー内の配列変数で管理するオブジェクト構造は以下の通り。

表 1 都道府県データスキーマ

プロパティ名	データ型	説明
<code>id</code>	Number	一意な識別子。登録時に自動採番（最大値 +1 等）。
<code>name</code>	String	都道府県名（例：大阪府）。
<code>capital</code>	String	県庁所在地（例：大阪市）。
<code>region</code>	String	地方区分（例：近畿）。

### 3 ディレクトリ・ページ構成

Express の標準的な構成に基づき、ビュー（EJS）ファイルを配置する。

```
project-root/
 |- app.js          (メインロジック・データ変数保持)
 |- public/          (CSS, 画像などの静的ファイル)
 |- views/          (EJS テンプレート)
   |- index.ejs     (一覧表示画面)
   |- show.ejs      (詳細表示画面)
   |- new.ejs       (新規登録フォーム)
   |- edit.ejs      (編集フォーム)
```

### 4 HTTP メソッドとルーティング

RESTful な設計に基づき、各 URL と HTTP メソッド、および対応する処理を定義する。※ブラウザの HTML フォーム標準仕様では PUT/DELETE が使用できないため、`method-override` ライブライアリを使用するか、POST メソッドで代用する想定とする（本仕様では論理的なメソッドを記載）。

表2 ルーティング一覧

機能	メソッド	パス (URL)	対応ビュー/処理
一覧表示	GET	/prefectures	views/index.ejs
新規作成フォーム	GET	/prefectures/new	views/new.ejs
詳細表示	GET	/prefectures/:id	views/show.ejs
編集フォーム	GET	/prefectures/:id/edit	views/edit.ejs
新規データ作成	POST	/prefectures	(処理後一覧ヘリダイレクト)
データ更新	PUT	/prefectures/:id	(処理後詳細ヘリダイレクト)
データ削除	DELETE	/prefectures/:id	(処理後一覧ヘリダイレクト)

## 5 ページ遷移図

画面間の遷移フローを以下に示す。

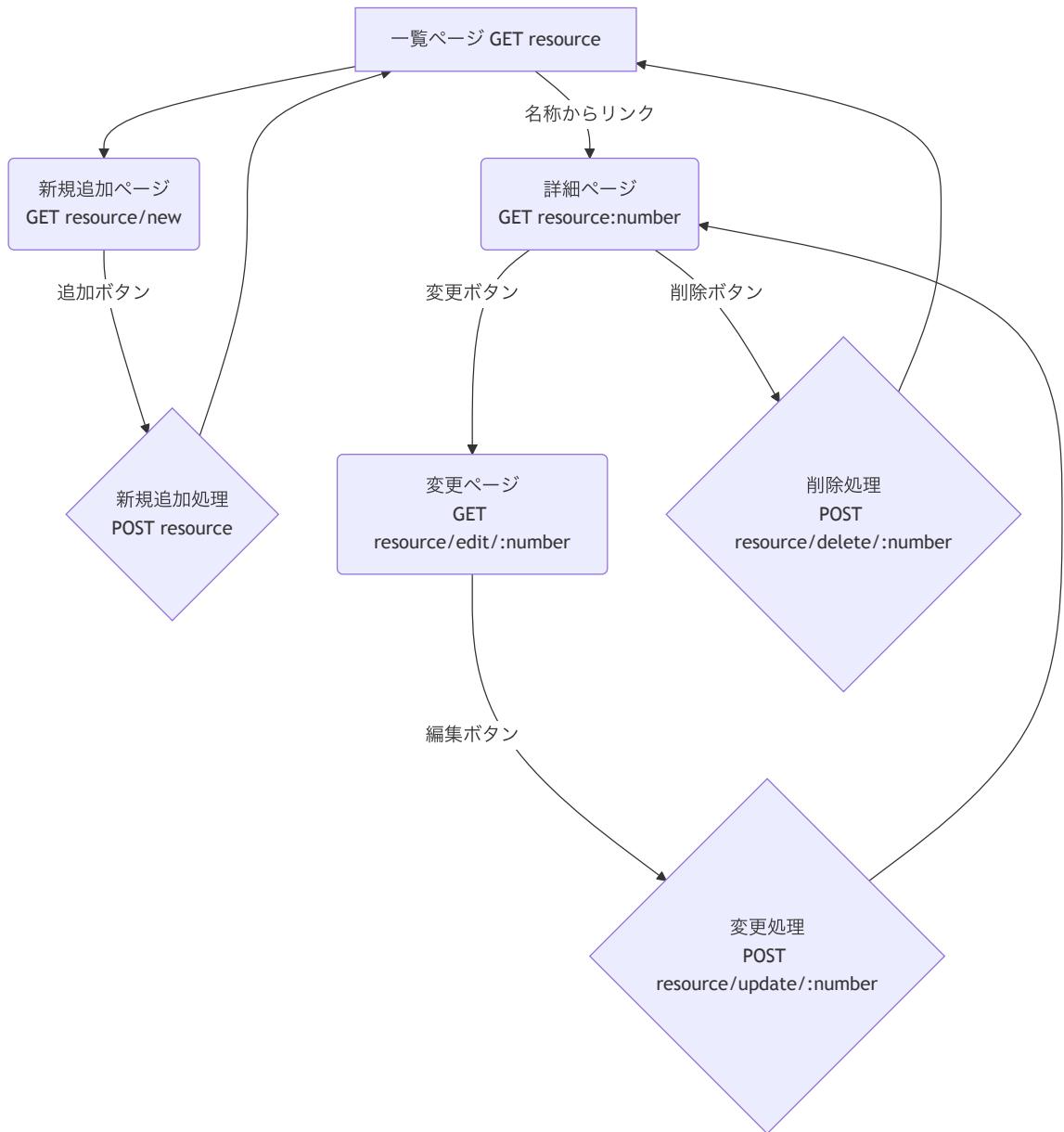


図 1 画面遷移フロー

## 6 各機能・リソース詳細

### 6.1 1. 一覧機能 (Index)

- **URL:** GET /prefectures
- **処理:** サーバー変数の全データを EJS に渡し, forEach 文を用いてテーブル形式でレンダリングする.
- **要素:** 「新規登録ボタン」, 各行ごとの「詳細リンク」.

### 6.2 2. 新規作成機能 (New / Create)

- **フォーム (GET /prefectures/new):**
  - 都道府県名, 県庁所在地, 地方の入力フィールドを表示.
  - 送信先は POST /prefectures.
- **作成処理 (POST /prefectures):**
  - 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.
  - 処理完了後, 一覧画面 (/prefectures) ヘリダイレクトする.

### 6.3 3. 詳細表示機能 (Show)

- **URL:** GET /prefectures/:id
- **処理:** URL パラメータの ID に基づき配列を検索 (find) し, 対象データを表示する.
- **要素:** 「編集ボタン」, 「削除ボタン」(form タグによる POST/DELETE 送信), 「一覧に戻るボタン」.

### 6.4 4. 編集・更新機能 (Edit / Update)

- **フォーム (GET /prefectures/:id/edit):**
  - 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.
  - 送信先は PUT /prefectures/:id.

- **更新処理 (PUT /prefectures/: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.
  - 詳細画面 (/prefectures/:id) ヘリダイレクトする.

## 6.5 5. 削除機能 (Destroy)

- **URL:** DELETE /prefectures/:id
- **処理:** ID に基づき配列から要素を取り除く (filter 等を使用).
- **挙動:** 削除完了後, 一覧画面 (/prefectures) ヘリダイレクトする.