

RouteEditor 機能仕様書

バージョン: 2.0 作成日: 2025年10月17日 プロジェクト名: RouteEditor

1. 概要

1.1 プロジェクト目的

RouteEditorは、国土地理院地図上でGeoJSONファイルの入出力と編集を行うWebアプリケーションです。地図上にポイント、ルート（中間点）、スポット（Polygon）を表示し、視覚的に確認しながらインタラクティブに編集できます。

1.2 主要機能

- **GeoJSONファイルの読み込み・表示・出力**（統計情報付き日付ファイル名）
- **ルート編集機能**（中間点の追加・移動・削除・最適化・クリア）
- **スポット編集機能**（スポットの追加・移動・削除・名称変更）
- **国土地理院地図タイル**による背景地図表示
- **フィーチャータイプ別スタイル表示**（円形・菱形・正方形）
- **インタラクティブな編集モード**（追加/移動/削除の切り替え）
- **自動ルート最適化**（最近傍貪欲法）
- **自動的に消えるトーストメッセージ通知**

1.3 技術スタック

- **フロントエンド**: Vanilla JavaScript（ES6モジュール）
- **地図ライブラリ**: Leaflet.js 1.9.4
- **地図タイル**: 国土地理院地図（標準地図）
- **ファイル形式**: GeoJSON（RFC 7946準拠）
- **座標系**: WGS84（EPSG:4326）

2. アプリケーション構成

2.1 ファイル構成

```
RouteEditor/
├── index.html          # メインHTMLファイル
├── styles.css          # スタイルシート
├── js/
│   ├── app.js         # メインアプリケーションロジック
│   ├── constants.js   # 設定定数（スタイル、モード定数）
│   ├── mapCore.js     # 地図初期化モジュール
│   ├── message.js     # メッセージ通知モジュール
│   ├── stats.js       # 統計情報管理モジュール
│   ├── fileIO.js      # ファイル入出力モジュール
│   └── routeEditor.js  # ルート編集機能モジュール
```

```
├── spotEditor.js      # スポット編集機能モジュール
├── docs/
│   ├── funcspec-202510.md # 機能仕様書（本ドキュメント）
│   ├── UsersGuide-202510.md # 利用者の手引
│   └── dataspec-geojson.md # GeoJSONデータ仕様書
└── CLAUDE.md          # プロジェクト概要（開発者向け）
```

2.2 モジュール設計

2.2.1 コアモジュール

- **app.js**: イベントハンドラー統合、モード切り替え処理
- **constants.js**: 定数管理（地図設定、スタイル、モード）
- **mapCore.js**: Leaflet地図の初期化とレイヤー管理

2.2.2 機能モジュール

- **message.js**: トーストメッセージ表示ロジック
- **stats.js**: 統計情報（ポイント数、ルート数、スポット数）計算
- **fileIO.js**: GeoJSONファイルの読み込みと出力処理

2.2.3 編集モジュール

- **routeEditor.js**: ルート編集の状態管理と編集機能
 - 状態管理（追加/移動/削除モード、選択ルート）
 - ルート中間点の追加・移動・削除
 - ルート最適化アルゴリズム（最近傍貪欲法）
 - ドロップダウン絞り込み機能
- **spotEditor.js**: スポット編集の状態管理と編集機能
 - スポットの追加・移動・削除
 - スポット名称変更
 - ドラッグ&ドロップによる位置変更

2.3 外部依存関係

- **Leaflet.js**: CDN経由で読み込み（1.9.4）
 - CSS: <https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.9.4/leaflet.css>
 - JS: <https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.9.4/leaflet.js>
- **国土地理院地図タイル**: <https://cyberjapandata.gsi.go.jp/xyz/std/{z}/{x}/{y}.png>

3. 機能詳細

3.1 地図表示機能

3.1.1 初期表示設定

- **初期中心座標**: 箕面大滝（北緯34.853667度、東経135.472041度）
- **初期ズームレベル**: 15

- **最大ズームレベル:** 18
- **地図タイル:** 国土地理院標準地図

3.1.2 地図コントロール

- **ズームコントロール:** 右下に配置
 - ズームイン (+) ボタン
 - ズームアウト (-) ボタン
- **スケールコントロール:** 右下に配置
 - メートル単位のスケール表示

3.1.3 地図操作

- マウスドラッグによるパン操作
 - マウスホイールによるズーム操作
 - ダブルクリックによるズームイン
 - ピンチ操作によるズーム (タッチデバイス)
-

3.2 モード切り替え機能

3.2.1 3つの編集モード

RouteEditorは3つの編集モードを持ち、ラジオボタンで切り替えます：

1. ファイルの入出力 (GeoJSON) モード

- GeoJSONファイルの読み込み・出力
- 統計情報の表示

2. ルートの位置編集モード

- ルート選択 (絞り込み機能付きドロップダウン)
- ルート中間点の追加・移動・削除
- ルート最適化 (中間点の並び替え)
- ルートクリア (全中間点削除)

3. スポットの位置編集モード

- スポット選択 (ドロップダウン)
- スポットの追加・移動
- スポット削除
- スポット名称変更

3.2.2 モード切り替え時の動作

- モード切り替え時、前のモードのアクティブな編集状態 (追加/移動/削除) を自動解除
 - 選択されたモードのパネルを表示、他のモードのパネルを非表示
 - スポットモードから離れる場合、スポットハイライトとドラッグ状態をリセット
-

3.3 GeoJSONファイル読み込み機能

3.3.1 ファイル選択

- **トリガー:** 「読み込み」 ボタンをクリック
- **対応形式:** `.geojson`, `.json`
- **ファイル選択方法:** ブラウザ標準のファイル選択ダイアログ
- **複数ファイル:** 非対応（1ファイルずつ読み込み、上書き）

3.3.2 データ読み込み処理

1. ファイルを選択
2. FileReader APIでファイル内容を読み込み
3. JSON.parse()でGeoJSONデータをパース
4. バリデーション（エラー時はエラーメッセージ表示）
5. 既存レイヤーとマーカーをクリア
6. GeoJSONデータを地図上に表示
 - ポイントGPS: 緑色円形マーカー
 - route_waypoint: 橙色菱形マーカー（初期非表示）
 - スポット: 青色正方形マーカー
7. ルート情報とスポット情報を抽出
8. ルート中間点マーカーを描画
9. 統計情報を更新
10. 地図範囲を自動調整（fitBounds）
11. 成功メッセージを表示（3秒間）

3.3.3 エラーハンドリング

- **JSONパースエラー:** 「ファイルの読み込みに失敗しました」 エラーメッセージ（6秒間表示）
 - **無効なGeoJSON形式:** 同上
 - **ファイル読み込みエラー:** 同上
-

3.4 GeoJSONファイル出力機能

3.4.1 出力トリガー

- **トリガー:** 「出力」 ボタンをクリック
- **前提条件:** GeoJSONファイルが読み込まれていること

3.4.2 ファイル名生成ルール

- **フォーマット:** `MapGPS-yyyymmdd_Pxx_Rxx_Sxx.geojson`
 - `yyyymmdd`: 日付（例: 20251017）
 - `Pxx`: ポイント数
 - `Rxx`: ルート数
 - `Sxx`: スポット数
- **例:** `MapGPS-20251017_P25_R3_S5.geojson`

3.4.3 保存方法

File System Access API対応ブラウザ（Chrome 86+、Edge 86+等）：

- `showSaveFilePicker()` APIを使用
- ユーザーが保存場所を選択可能
- 拡張子フィルタ: `.geojson`, `.json`

非対応ブラウザ（Firefox、Safari等）：

- 従来のダウンロード方式（Blob + `<a> download`属性）
- ブラウザのデフォルトダウンロードフォルダに保存

3.4.4 エラーハンドリング

- **データ未読み込み**: 「出力するデータがありません」警告メッセージ（4.5秒間表示）
 - **保存キャンセル**: メッセージなし（処理中断）
 - **保存失敗**: エラーメッセージ（6秒間表示）
-

3.5 ルート編集機能

3.5.1 ルート選択機能

3段階絞り込みドロップダウン:

1. 開始点の頭文字絞り込み (`routeStart`)

- ルートの開始点または終了点のID頭文字で絞り込み
- 例: "T" を選択すると "T" で始まるポイントを含むルートのみ表示

2. ポイントID絞り込み (`routeEnd`)

- 開始点の頭文字で絞り込まれたポイントIDのリスト
- 例: "T15" を選択すると "T15" を含むルートのみ表示

3. ルート選択 (`routePath`)

- 絞り込まれたルート of リスト
- フォーマット: 開始ポイント ~ 終了ポイント (中間点数)
- 例: T15 ~ T16 (5)

リセット機能:

- 「リ」 ボタンをクリックで全ドロップダウンをリセット
- ルートハイライトも解除

ルート選択変更時の動作:

- アクティブな編集モード（追加/移動/削除）を自動解除
- 選択されたルートをハイライト表示
 - 開始・終了ポイント: 赤色 (`#ff0000`)

- ルート中間点: 赤色 (#ef454a)
- ルート線: 赤色 (#ef454a)、線幅2px

3.5.2 中間点追加機能

トリガー: 「追加」 ボタンをクリック

動作フロー:

1. ルート選択確認（未選択の場合は警告）
2. 既に追加モードの場合は解除
3. 他のモード（移動/削除）が有効な場合は解除
4. 追加モードを開始
 - ボタンに **active** クラス追加（青色背景）
 - 地図カーソルを十字に変更
 - メッセージ表示: 「地図上をクリックして中間点を追加してください」
5. 地図クリック時に中間点を追加
 - クリック位置に新しい中間点を作成
 - **waypoint_number** を自動採番（最大値+1）
 - GeoJSONデータに追加
 - 地図上にマーカー表示（赤色菱形）
 - ルートを自動最適化（メッセージなし）
 - ルート線を再描画

解除方法:

- 「追加」 ボタンを再度クリック
- 他のモードボタンをクリック
- ルート選択を変更
- 別のモードに切り替え

3.5.3 中間点移動機能

トリガー: 「移動」 ボタンをクリック

動作フロー:

1. ルート選択確認（未選択の場合は警告）
2. 既に移動モードの場合は解除
3. 他のモード（追加/削除）が有効な場合は解除
4. 移動モードを開始
 - ボタンに **active** クラス追加（青色背景）
 - 中間点マーカーをクリック可能に設定（カーソル: pointer）
 - メッセージ表示: 「移動する中間点をクリックしてください」
5. 中間点クリック時
 - クリックされた中間点をドラッグ可能に設定（カーソル: move）
 - 他のドラッグ可能マーカーを無効化
6. ドラッグ中
 - リアルタイムで座標を更新

- ルート線を即座に再描画
- 7. ドラッグ終了時
 - 座標を確定
 - ルートを自動最適化（メッセージなし）
 - ルート線を再描画
 - 中間点を再度クリック可能に設定

解除方法:

- 「移動」ボタンを再度クリック
- 他のモードボタンをクリック
- ルート選択を変更
- 別のモードに切り替え

3.5.4 中間点削除機能

トリガー:「削除」ボタンをクリック

動作フロー:

1. ルート選択確認（未選択の場合は警告）
2. 既に削除モードの場合は解除
3. 他のモード（追加/移動）が有効な場合は解除
4. 削除モードを開始
 - ボタンに **active** クラス追加（青色背景）
 - 中間点マーカーをクリック可能に設定（カーソル: pointer）
 - メッセージ表示: 「中間点をクリックして削除できます」
5. 中間点クリック時
 - GeoJSONデータから削除
 - 地図からマーカーを削除
 - ルートを自動最適化（メッセージなし）
 - ルート線を再描画
 - 削除モードは継続（複数削除可能）

解除方法:

- 「削除」ボタンを再度クリック
- 他のモードボタンをクリック
- ルート選択を変更
- 別のモードに切り替え

3.5.5 ルート最適化機能

トリガー:「最適化」ボタンをクリック

アルゴリズム: 最近傍貪欲法（Nearest Neighbor Greedy Algorithm）

動作フロー:

1. ルート選択確認（未選択の場合は警告）

2. アクティブなモードを解除
3. 開始ポイントと終了ポイントを取得
4. ルートの全中間点を取得
5. 最適化処理:
 - 開始ポイントから最も近い中間点を選択
 - 選択された中間点から次の最も近い中間点を選択
 - すべての中間点を選択されるまで繰り返し
6. `waypoint_number` を再採番（1から順番に）
7. 中間点マーカーを再描画
8. ルート線を再描画
9. 成功メッセージ表示: 「ルートを最適化しました（xx個の中間点）」

自動最適化: 以下の操作後、自動的に最適化が実行されます（メッセージなし）：

- 中間点追加後
- 中間点削除後
- 中間点移動（ドラッグ）終了後

3.5.6 ルートクリア機能

トリガー: 「クリア」 ボタンをクリック

動作フロー:

1. ルート選択確認（未選択の場合は警告）
 2. アクティブなモードを解除
 3. 確認ダイアログ表示: 「ルート [ルート名] を削除しますか？」
 4. 確認後の処理:
 - GeoJSONデータから全中間点を削除
 - 地図から全中間点マーカーを削除
 - ルート線を削除
 - 開始・終了ポイントのマーカー色を元に戻す（緑色）
 - `allRoutes` 配列から削除
 - ドロップダウンを更新
 - ルート選択をクリア
 5. 成功メッセージ表示: 「ルートを削除(=クリア)しました」
-

3.6 スポット編集機能

3.6.1 スポット選択機能

ドロップダウン選択 (`spotSelect`):

- スポット一覧をドロップダウンで表示
- フォーマット: スポット名
- 選択時にスポットをハイライト表示
 - Point型スポット: シアン色（`#00ffff`）の正方形
 - Polygon型スポット: シアン色（`#00ffff`）の塗りつぶし

スポット数表示:

- スポット数を専用フィールドに表示

スポット名表示・編集:

- 選択されたスポットの名前を表示
- テキストボックスで直接編集可能
- フォーカスを外すと自動保存
- 成功メッセージ表示: 「スポット名を更新しました」

3.6.2 スポット追加・移動機能

トリガー: 「追加・移動」 ボタンをクリック

動作フロー（追加モード）:

1. データ読み込み確認（未読み込みの場合は警告）
2. 追加・移動モードを開始
 - ボタンに **active** クラス追加（青色背景）
 - 地図カーソルを十字に変更
 - メッセージ表示: 「地図上をクリックして新しいスポットを追加してください」
3. 地図クリック時に新しいスポットを追加
 - クリック位置に新しいスポットを作成
 - スポット名は自動生成（**仮1**, **仮2**, ...、重複回避）
 - GeoJSONデータに追加
 - 地図上にマーカー表示（青色正方形）
 - スポットドロップダウンを更新
 - 新しいスポットを自動選択してハイライト

動作フロー（移動モード）:

1. スポットが選択されている場合
 - 選択されたスポットマーカーをドラッグ可能に設定
 - メッセージ表示: 「スポットをドラッグして移動できます」
2. ドラッグ中
 - リアルタイムで座標を更新
3. ドラッグ終了時
 - 座標を確定
 - 成功メッセージ表示: 「スポットの位置を更新しました」

解除方法:

- 「追加・移動」 ボタンを再度クリック
- 別のモードに切り替え

3.6.3 スポット削除機能

トリガー: 「削除」 ボタンをクリック

動作フロー:

1. スポット選択確認（未選択の場合は警告）
 2. アクティブなモード（追加・移動）を解除
 3. 確認ダイアログ表示: 「スポット「[スポット名]」を削除しますか？」
 4. 確認後の処理:
 - GeoJSONデータから削除
 - 地図からマーカーを削除
 - `spotMarkerMap` から削除
 - `allSpots` 配列から削除
 - 選択状態をリセット
 - ドロップダウンと統計を更新
 5. 成功メッセージ表示: 「スポットを削除しました」
-

3.7 フィーチャー表示機能

3.7.1 対応ジオメトリタイプ

- **Point:** 単一ポイント
- **LineString:** 単一ルート（現在未使用）
- **MultiLineString:** 複数ルート（現在未使用）
- **Polygon:** 単一エリア
- **MultiPolygon:** 複数エリア

3.7.2 フィーチャータイプ別スタイル

ポイントGPS (`type: "ポイントGPS"`) :

- **形状:** 円形 (CircleMarker)
- **色:** 緑色 (`#008000`)
- **サイズ:** 半径6ピクセル
- **枠線:** なし
- **不透明度:** 100%
- **用途:** ルートの開始・終了地点

ルート中間点 (`type: "route_waypoint"`) :

- **形状:** 菱形 (ダイヤモンド型、45度回転)
- **色:** 橙色 (`#f58220`)、選択時は赤色 (`#ef454a`)
- **サイズ:** 8x8ピクセル
- **枠線:** なし
- **不透明度:** 80%
- **用途:** ルート上の中間地点

スポット (`type: "spot"`) :

- **形状:** 正方形 (Point型) またはポリゴン (Polygon型)
- **色:** 青色 (`#0000ff`)、選択時はシアン色 (`#00ffff`)
- **サイズ:** 12x12ピクセル (Point型)
- **枠線:** なし

- 不透明度: 80%
- 用途: 休憩所や施設地点

ライン・ポリゴン（LineString, Polygon等）:

- 色: 青色（#3388ff）
- 線幅: 3ピクセル
- 線の不透明度: 80%
- 塗りつぶし不透明度: 30%

ルート線（選択時）:

- 色: 赤色（#ef454a）
- 線幅: 2ピクセル

3.7.3 ポップアップ機能

- フィーチャーをクリックすると `properties.name` または `properties.id` を表示
- ポップアップ内容: フィーチャー名のみ

3.8 統計情報表示機能

3.8.1 統計項目

制御パネル内に以下の統計を表示:

項目	単位	カウント方法
ファイル	-	読み込まれたファイル数（常に0または1）
ポイント	点	<code>type: "ポイントGPS"</code> のPointフィーチャー数
ルート	本	<code>route_waypoint</code> の <code>route_id</code> ユニーク数
スポット	個	<code>type: "spot"</code> のPointフィーチャー数 + Polygon/MultiPolygonの数

3.8.2 カウントロジック

ポイント数:

- `geometry.type === "Point"` かつ `properties.type === "ポイントGPS"`

ルート数:

- `route_waypoint` の `route_id` プロパティのユニーク数
- `route_id`形式: `route_[開始ID]_to_[終了ID]`

スポット数:

- `geometry.type === "Point"` かつ `properties.type === "spot"`
- または `geometry.type === "Polygon" / "MultiPolygon"`

3.8.3 表示更新タイミング

- ファイル読み込み成功時
- スポット追加・削除時
- ファイル読み込みエラー時（0にリセット）

3.9 メッセージ通知機能

3.9.1 メッセージタイプ

- **success**: 成功メッセージ（薄い水色背景 #d1ecf1）
- **warning**: 警告メッセージ（黄色背景 #fffc107）
- **error**: エラーメッセージ（赤色背景 #dc3545）

3.9.2 表示仕様

- **表示位置**: 画面中央
- **表示時間**:
 - success: 3秒間
 - warning: 4.5秒間
 - error: 6秒間
- **アニメーション**:
 - 表示: フェードイン + スケールアップ（0.3秒）
 - 消去: フェードアウト（0.3秒）
- **スタイル**:
 - パディング: 12px 24px
 - 角丸: 6px
 - ドロップシャドウ: 0 4px 12px rgba(0,0,0,0.3)
 - z-index: 10000

3.9.3 メッセージ一覧

ファイル入出力:

トリガー	メッセージ	タイプ
ファイル読み込み成功	GeoJSONファイルを読み込みました	success
ファイル読み込み失敗	ファイルの読み込みに失敗しました: [エラー詳細]	error
ファイル出力成功	GeoJSONファイルを出力しました	success
出力時データ未読み込み	出力するデータがありません。先にGeoJSONファイルを読み込んでください。	warning

ルート編集:

トリガー	メッセージ	タイプ
ルート未選択時の操作	ルートを選択してください	warning
ルート選択変更時（追加モード）	ルート選択変更により追加モードを解除しました	success
ルート選択変更時（移動モード）	ルート選択変更により移動モードを解除しました	success
ルート選択変更時（削除モード）	ルート選択変更により削除モードを解除しました	success
追加モード解除	追加モードを解除しました	success
移動モード解除	移動モードを解除しました	success
削除モード解除	削除モードを解除しました	success
中間点追加成功	中間点を追加しました	success
最適化成功	ルートを最適化しました（xx個の中間点）	success
ルートクリア成功	ルートを削除(=クリア)しました	success

スポット編集:

トリガー	メッセージ	タイプ
データ未読み込み時	先にGeoJSONファイルを読み込んでください	warning
スポット未選択時の削除	削除するスポットを選択してください	warning
スポット追加成功	スポットを追加しました	success
スポット名更新成功	スポット名を更新しました	success
スポット移動成功	スポットの位置を更新しました	success
スポット削除成功	スポットを削除しました	success
追加・移動モード解除	追加・移動モードを解除しました	success

4. UI仕様

4.1 レイアウト

- **地図エリア**: 画面全体（100vh x 100vw）
- **制御パネル**: 右上固定配置（絶対位置）

4.2 制御パネル仕様

- **背景色**: 白（#ffffff）
- **枠線**: 2px solid #ccc
- **角丸**: 8px
- **パディング**: 20px
- **最小幅**: 240px
- **ドロップシャドウ**: 0 2px 10px rgba(0,0,0,0.1)

- **z-index:** 1000

4.3 モード選択部

- **ラジオボタン:** 3つのモード切り替え
- **選択状態スタイル:** 青色 (`#007bff`)、太字 (`font-weight: bold`)

4.4 GeoJSONパネル仕様

- **背景色:** 薄いグレー (`#f8f9fa`)
- **枠線:** 2px solid `#666`
- **角丸:** 8px
- **パディング:** 15px
- **レイアウト:** 左右2カラム構成
 - 左: ボタン (読み込み、出力)
 - 右: 統計情報

4.5 ルートパネル仕様

- **背景色:** 薄いグレー (`#f8f9fa`)
- **枠線:** 2px solid `#666`
- **角丸:** 8px
- **パディング:** 15px
- **ドロップダウン:** 3段階絞り込み (開始点、ポイント、ルート)
- **ボタン:** 5つ (追加、移動、削除、最適化、クリア) × 2行

4.6 スポットパネル仕様

- **背景色:** 薄いグレー (`#f8f9fa`)
- **枠線:** 2px solid `#666`
- **角丸:** 8px
- **パディング:** 15px
- **ドロップダウン:** スポット選択
- **テキストボックス:** スポット名編集
- **ボタン:** 2つ (追加・移動、削除)

4.7 ボタンスタイル

- **パディング:** 8px 16px
- **枠線:** 2px solid `#666`
- **角丸:** 4px
- **背景色:** 白
- **ホバー時:** 薄いグレー背景 (`#f0f0f0`)
- **アクティブ時:** 青色背景 (`#007bff`)、白色テキスト

4.8 統計表示スタイル

- **入力欄幅:** 32px
- **テキスト配置:** 右寄せ

- 背景色: #f8f9fa
- 読み取り専用: readonly属性

5. データ仕様

5.1 GeoJSON形式

詳細は [docs/dataspec-geojson.md](#) を参照。

5.1.1 基本構造

```
{
  "type": "FeatureCollection",
  "features": [...]
}
```

5.1.2 座標系

- 座標系: WGS84 (EPSG:4326)
- 形式: [経度, 緯度, 標高 (オプション)]
- 精度: 小数点以下5桁 (約1m精度)

5.1.3 対応フィーチャータイプ

- **ポイントGPS** (`type: "ポイントGPS"`): ルートの開始・終了地点
 - 必須プロパティ: `id`, `type`
- **ルート中間点** (`type: "route_waypoint"`): ルート上の中間地点
 - 必須プロパティ: `type`, `route_id`, `waypoint_number`
 - `route_id`形式: `route_[開始ID]_to_[終了ID]`
- **スポット** (`type: "spot"`): 休憩所や施設地点
 - 必須プロパティ: `type`, `name`

6. ブラウザ対応

6.1 必須機能

- ES6モジュール対応
- FileReader API対応
- Leaflet.js対応 (IE11+)
- DOM Manipulation (`querySelector`, `addEventListener`等)

6.2 オプション機能

- File System Access API対応 (Chrome 86+、Edge 86+)
 - 非対応ブラウザでは従来のダウンロード方式にフォールバック

6.3 推奨ブラウザ

- Google Chrome 90+
- Microsoft Edge 90+
- Firefox 89+
- Safari 14+

7. パフォーマンス考慮事項

7.1 制約事項

- 大容量GeoJSONファイル（数千フィーチャー以上）でパフォーマンス低下の可能性
- すべての処理はクライアントサイドで実行
- メモリ使用量はデータサイズに比例

7.2 最適化

- GeoJSONレイヤーグループによる効率的なレイヤー管理
- fitBounds時のパディング設定（10px）
- 不要なレイヤーの自動クリア（新規読み込み時）
- ルート最適化アルゴリズムの実装（最近傍貪欲法）
- イベントハンドラーの適切な解除（メモリリーク防止）

8. 開発・デバッグ

8.1 ローカルサーバー起動

ES6モジュール使用のため、CORS制限回避にローカルHTTPサーバーが必須:

```
# Python使用
python -m http.server 8000

# または npx使用
npx serve .
```

アクセス: <http://localhost:8000>

8.2 デバッグ方法

- ブラウザ開発者ツールのコンソールでエラー確認
- `console.warn()` でFile System Access APIエラーをログ出力
- グローバル変数 `geoJsonLayer` で読み込み状態確認
- モジュールごとの状態管理オブジェクト（`routeEditor.state`, `spotEditor.allSpots`等）

8.3 モジュール構成

すべてのJavaScriptファイルはES6モジュールとして実装されています：

- `import/export` 構文使用
- `<script type="module">` でメインスクリプト (app.js) を読み込み
- 各モジュールは単一責任の原則に従って設計

9. アルゴリズム詳細

9.1 ルート最適化アルゴリズム

アルゴリズム名: 最近傍貪欲法 (Nearest Neighbor Greedy Algorithm)

目的: ルート上の中間点を、開始ポイントから終了ポイントまで最短経路に近い順序に並び替える

手順:

1. 開始ポイントの座標を取得
2. 終了ポイントの座標を取得
3. 全ルート中間点を取得
4. 現在位置を開始ポイントに設定
5. 未訪問の中間点がある間、以下を繰り返し:
 - 現在位置から最も近い中間点を選択 (ハバーサイン公式で距離計算)
 - 選択された中間点を訪問済みリストに追加
 - 現在位置を選択された中間点に更新
6. 訪問順に `waypoint_number` を1から採番
7. 中間点マーカーを再描画

距離計算: ハバーサイン公式 (Haversine Formula)

- 地球を球体と仮定した2点間の大圏距離計算
- 精度: 数メートル程度
- 計算式:

```
const R = 6371; // 地球の半径 (km)
const dLat = (lat2 - lat1) * Math.PI / 180;
const dLng = (lng2 - lng1) * Math.PI / 180;
const a = Math.sin(dLat / 2) * Math.sin(dLat / 2) +
          Math.cos(lat1 * Math.PI / 180) * Math.cos(lat2 * Math.PI / 180) *
          Math.sin(dLng / 2) * Math.sin(dLng / 2);
const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
const distance = R * c;
```

計算量: $O(n^2)$ (n = 中間点数)

制約:

- 局所最適解であり、必ずしも最短経路ではない
- 中間点数が多い場合 (100点以上)、処理時間が増加する可能性

10. 変更履歴

バージョン	日付	変更内容
1.0	2025-09-30	初版作成
2.0	2025-10-17	ルート編集機能実装、スポット編集機能実装、モジュール構成リファクタリング

11. 参照ドキュメント

- [GeoJSONデータ仕様書](#)
- [利用者の手引](#)
- [プロジェクト概要（開発者向け）](#)
- [Leaflet.js公式ドキュメント](#)
- [国土地理院地図タイル](#)
- [RFC 7946 \(GeoJSON仕様\)](#)