

RouteEditor 機能仕様書

バージョン: 1.0 作成日: 2025年09月30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（LineString）、スポット（Polygon）を表示し、視覚的に確認しながらGeoJSONファイルの管理を行います。

1.2 主要機能

- GeoJSONファイルの読み込み・表示
- GeoJSONファイルの出力（日付付きファイル名）
- 国土地理院地図タイルによる背景地図表示
- フィーチャタイプ別のマーカースタイル表示
- 統計情報の表示（ファイル数、ポイント数、ルート数、スポット数）
- 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  
├── docs/  
│   ├── funcspec-202509.md # 機能仕様書（本ドキュメント）  
│   ├── UsersGuide-202509.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 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 GeoJSONファイル読み込み機能

3.2.1 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)

3.2.2 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. 地図上にデータを表示
6. 統計情報を更新
7. 地図範囲を自動調整（fitBounds）
8. 成功メッセージを表示（3秒間）

3.2.3 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
-

3.3 GeoJSONファイル出力機能

3.3.1 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

3.3.2 ファイル名生成ルール

- **フォーマット**: MapGPS-yyyyymmdd.geojson
- **日付取得**: システム日付（実行時点）
- **例**: MapGPS-20250930.geojson

3.3.3 保存方法

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

- `showSaveFilePicker()` APIを使用
- 読み込んだフォルダを保存先として提案（ブラウザが記憶）
- 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

3.3.4 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）
-

3.4 フィーチャー表示機能

3.4.1 対応ジオメトリタイプ

- **Point**: 単一ポイント
- **LineString**: 単一ルート
- **MultiLineString**: 複数ルート
- **Polygon**: 単一エリア
- **MultiPolygon**: 複数エリア

3.4.2 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%

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)
- **サイズ**: 8x8ピクセル
- **枠線**: なし
- **不透明度**: 80%

スポット (**type**: "spot") :

- **形状**: 正方形
- **色**: 青色 (#0000ff)
- **サイズ**: 12x12ピクセル
- **枠線**: なし
- **不透明度**: 80%

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%

3.4.3 ポップアップ機能

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

3.5 統計情報表示機能

3.5.1 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	LineString/MultiLineStringの本数、または <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

3.5.2 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"`
- `type` が未指定の場合もポイントとしてカウント

ルート数:

- 優先順位1: LineString/MultiLineStringの総数
- 優先順位2: `route_waypoint` の `route_id` プロパティのユニーク数

スポット数:

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

3.5.3 表示更新タイミング

- ファイル読み込み成功時
- ファイル読み込みエラー時（0にリセット）

3.6 メッセージ通知機能

3.6.1 メッセージタイプ

- **success:** 成功メッセージ（緑背景 → 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`）

3.6.2 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

3.6.3 メッセージ一覧

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

3.7 モード切り替え機能（UI準備済み、機能未実装）

3.7.1 モード一覧

1. **ファイルの入出力（GeoJSON）**: 現在実装済み
2. **ルート的位置編集**: UI準備済み、機能未実装
3. **スポット的位置編集**: UI準備済み、機能未実装

3.7.2 現在の動作

- ラジオボタンで選択可能
- 選択されたモードのラベルが青色・太字で表示
- 実際の機能切り替えは未実装

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 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

4.4 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)

4.5 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입

- ポイントGPS (`type: "ポイントGPS"`)
 - ルート中間点 (`type: "route_waypoint"`)
 - スポット (`type: "spot"`)
-

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+)

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)
-

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`, `loadedData` で読み込み状態確認

9. 今後の拡張予定

9.1 未実装機能

1. ルートの位置編集モード

- インタラクティブなルート描画・編集
- Leaflet.drawプラグイン導入検討

2. スポットの位置編集モード

- インタラクティブなポリゴン作成・編集
- 新規スポット追加機能

9.2 拡張方針

- 新しいモジュールファイルの作成（例: `js/route-editor.js`）
- `constants.js` に必要な設定値を追加
- `app.js` でモジュールをインポートし初期化

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

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](#)

