

# PointMarker 利用者の手引

Version 2025年9月版 v5.0

## はじめに

PointMarkerは、ハイキングマップの画像からポイント・スポット・ルートをマーキングし、座標データを管理するWebアプリケーションです。登山計画の作成や、実際に歩いたルートの記録、さらに特徴的なスポット情報の管理にも活用できます。

v5.0の主要改善点：

- リファクタリングによる大幅なパフォーマンス向上
- 統合オブジェクト検出システムの実装
- 統一バリデーション・スタイル管理機能
- Favicon対応（♀ピンアイコン）
- より直感的で安定した操作性

すべての処理はブラウザ内で完結し、プライバシーを保護しながら安全にご利用いただけます。

## システム要件

### 対応ブラウザ

- 推奨**：Chrome 86以降（File System Access API対応）、Edge 86以降
- 対応**：Firefox 最新版、Safari 14以降（基本機能利用可能）
- 必要な機能**：HTML5 Canvas、FileReader API、ES6モジュール対応

### 重要：ローカルサーバーの起動

ES6モジュールを使用しているため、**必ずローカルサーバー経由**でアクセスしてください：

```
# Python (推奨)
python -m http.server 8000

# または Node.js
npx serve .
```

その後ブラウザで <http://localhost:8000> を開いてください。

**注意**：ファイルを直接ブラウザで開く（file://）とCORSエラーで動作しません。

## ファイル形式

- 入力画像**：PNG形式のみ
- データ保存**：JSON形式

## ネットワーク

- ・ インターネット接続不要（ローカル処理のみ）
- ・ すべてのデータはローカルファイルに保存
- ・ 外部サーバーへのデータ送信は一切なし

## 新機能：Favicon対応

- ・ ブラウザタブに ♀ ピンアイコンが表示されます
- ・ 404エラー（favicon.ico not found）を解消
- ・ PointMarkerの視覚的アイデンティティを強化

## 基本的な使い方

### 1. アプリケーションの起動

1. ローカルサーバーを起動（上記参照）
2. ブラウザで <http://localhost:8000> にアクセス
3. PointMarkerのメイン画面が表示されることを確認
4. ブラウザタブに ♀ ピンアイコンが表示されることを確認

### 2. 地図画像の読み込み

1. \*\*「PNG画像を選択」\*\*ボタンをクリック
2. ハイキングマップのPNG画像ファイルを選択
3. 画像が自動的にキャンバスに表示されることを確認

### File System Access API対応ブラウザ（Chrome/Edge）の場合：

- ・ ファイル選択ダイアログが表示されます
- ・ 保存時にファイル名・保存場所を指定可能

### 注意事項：

- ・ PNG形式の画像のみ対応（JPEG、GIF等は非対応）
- ・ 画像は表示領域に合わせて自動スケーリングされます
- ・ アスペクト比は保持されます
- ・ リサイズ処理が最適化され、より滑らかな動作を実現

### 3. レイアウトの選択

画面上部で2つのレイアウトから選択できます：

- ・ **サイドバー**（推奨）：地図とコントロールパネルが横並び表示
- ・ **オーバーレイ**：コントロールパネルが地図上に半透明で重ね表示

### 4. 編集モードの選択

3つの編集モードから用途に応じて選択できます：

- ・ **ポイント編集**：地図上の重要ポイント（山頂、山小屋、分岐点など）を管理
- ・ **ルート編集**：歩行ルートの詳細記録・計画作成
- ・ **スポット編集**：観光スポット、景勝地、特徴的な場所の記録

## ポイント編集機能

### ポイントの配置

1. \*\*「ポイント編集」\*\*モードを選択（デフォルト）
2. 地図上の任意の場所をクリック
3. 赤い円マーカーが表示され、**動的ポップアップ入力ボックス**が自動表示
4. **自動フォーカス**：新規ポイント作成時、カーソルが末尾に自動設定
5. ポイントIDを入力（X-nn形式：例 A-01、J-12）

### ポイントID管理

- **形式**: X-nn（英大文字1桁-数字2桁）例：A-01, B-15, Z-99
- **自動補正**: 統一されたValidatorsクラスによる処理
  - 全角→半角変換、小文字→大文字変換、0埋め処理
- **入力制御**:
  - 入力中（typing）：補正処理なし、リアルタイム表示
  - フォーカス離脱時（blur）：自動補正実行

### ポイントの編集・移動・削除

- **編集**: 既存ポイントクリック→対応する入力フィールドにフォーカス
- **移動**: ドラッグ&ドロップによる位置変更（ポイント編集モードのみ）
  - 統合ドラッグ処理により、より滑らかで安定した移動を実現
  - オブジェクト検出が改善され、正確なドラッグ開始が可能
- **削除**: Escapeキー、または空入力でblur

### ポイント編集時のUI表示制御

- **ポイントIDポップアップ**: 表示・編集可能
- **カーソル表示**: crosshairカーソルで一貫した操作感
- **統一スタイリング**: 統一された関数による視覚的フィードバック

## スポット編集機能（改良版）

### スポットとは

観光スポット、景勝地、展望台、滝、特徴的な地形など、地図上の「場所」を四角形マーカー（■）で記録する機能です。ポイントがID管理による体系的な地点管理に対し、スポットは名前による直感的な場所管理に適しています。

### スポットの配置

1. \*\*「スポット編集」\*\*モードを選択
2. 地図上の任意の場所をクリック
3. \*\*四角形マーカー（■）\*\*が青色で表示され、**動的スポット名入力ボックス**が自動表示
4. **自動フォーカス**：新規スポット作成時、カーソルが末尾に自動設定
5. スポット名を入力（例：「展望台」「白山神社」「清水の滝」）

### スポット名管理

- **入力制御**: 最大10文字制限、自由な名前設定
- **リアルタイム表示**: 文字入力中の即座反映
- **trim処理**: 空白文字の自動除去
- **統合管理**: SpotManagerクラスによる効率的なデータ管理

## スポットの編集・移動・削除

- **編集**: 既存スポット (■) クリック→対応する入力フィールドにフォーカス
- **移動**: 統合ドラッグ処理による位置変更 (スポット編集モードのみ)
- **削除**: Escapeキー、または空入力でblur

## スポット編集時のUI表示制御

- **ポイントIDポップアップ**: 完全に非表示 (他機能との区別)
- **スポット入力ボックス**: 表示・編集可能
- **カーソル表示**: crosshairカーソルで統一された操作感

## スポットの視覚表示 (更新)

- **マーカー形状**: 四角形 (■)
- **色**: 青色系 (#0066ff、ポイントの赤色と明確に区別)
- **サイズ**: ポイントより大きめ (12px、視認性重視)
- **描画処理**: CanvasRenderer.drawSquare()による最適化された描画

## スポット一括操作

- 「**スポットをクリア**」: すべてのスポットを即座に削除 (赤色ボタン)
- 「**スポットをJSON出力**」: スpotトデータをJSON形式で保存 (緑色ボタン)
- 「**スポットのJSON読込**」: 既存スポットJSONからデータ復元

## ルート編集機能

### ルート編集の準備

1. 事前にポイント編集でルートの開始・終了点を配置
2. \*\*「ルート編集」\*\*モードに切り替え
3. **UI制御** : 既存ポイント・スポットの編集が自動的に制限されます

### ルート編集時のUI変更

#### 完全な視覚的制御 :

- **ポイントIDポップアップ**: 表示 (灰色背景で無効化表示)
- **スポット入力ボックス**: 完全に非表示
- **通常ポイント**: 統一スタイル関数による灰色表示
- **開始・終了ポイント**: 完全白背景+青枠線で強調表示
- **無効化表示**: 全入力フィールドがdisabled状態

### 開始・終了ポイント入力とバリデーション (強化版)

## 統合バリデーション機能 :

1. **形式チェック**: Validatorsクラスによる厳密なX-nn形式検証
2. **存在チェック**: 既存ポイントIDとの照合
  - 存在しないポイント → 統一エラー表示関数による赤枠エラー
  - 詳細ツールチップでエラー理由説明
3. **重複チェック**: 開始・終了ポイント同一値検出
  - 重複時 → **両フィールド同時に赤枠表示** (強化された検証)
  - 統一メッセージによる注意喚起

## 統一エラーフィードバック :

- **形式エラー**: 薄いピンク背景 + 赤枠 + 「X-nn形式で入力してください」
- **存在エラー**: 赤枠のみ + 「ポイントID「〇〇」は存在しません」
- **重複エラー**: 両フィールド赤枠 + 「開始ポイントと終了ポイントは異なるポイントIDを指定してください」

## ルートの作成手順

1. **開始ポイント**欄に開始点のID（例：A-01）を入力
2. **終了ポイント**欄に終了点のID（例：B-05）を入力
3. **自動バリデーション** : 統合検証システムによる即座の検証
4. **視覚フィードバック** : 指定されたポイントが自動的に白背景+青枠で強調
5. 地図上でルートの中間点をクリックして順次配置
6. 橙色の菱形で中間点が表示され、ルートラインで接続

## ルート情報表示

- **中間点数**: 開始・終了ポイント入力エリア内に表示（終了ポイント入力フィールドの直下）
- **リアルタイム更新**: 中間点追加・削除に応じて自動更新

## ルートデータの操作

- **「ルートをクリア」**: 全中間点と開始・終了ポイント設定を削除（赤色ボタン）
- **「ルートをJSON出力」**:
  - 統合検証システムによる総合チェック→エラーがあれば詳細メッセージ表示
  - ルートデータをJSON形式で保存（緑色ボタン）
- **「ルートのJSON読み込み」**: 既存ルートJSONからデータ復元・UI再構築

## ファイル操作 (改良版)

### 自動ファイル名生成

保存時のファイル名は自動生成されます :

- **ポイント**: {画像名}\_points.json
- **ルート**: {画像名}\_route\_{開始ポイント}\_to\_{終了ポイント}.json
- **スポット**: {画像名}\_spots.json

例 :

- `hakusan_points.json`
- `hakusan_route_A-01_to_B-05.json`
- `hakusan_spots.json`

高度な保存機能 (Chrome/Edge)

**File System Access API**対応ブラウザでは：

- ファイル名・保存場所の任意指定
- 既存ファイルの上書き確認
- より直感的なファイル操作
- FileHandlerクラスによる統合されたファイル処理

**非対応ブラウザ (Firefox/Safari) :**

- 従来のダウンロード機能で保存
- ブラウザのダウンロードフォルダに自動保存

読み込み機能

1. 対応するJSONファイルを選択
2. **座標変換** : CoordinateUtilsクラスによる精密な座標変換
3. **UI再構築** : 入力ボックス・状態表示も完全復元
4. **エラーハンドリング**: 統合されたエラー処理による安全な読み込み

画面の見方

メイン画面の構成

- **キャンバス領域**: 地図画像とマーカーの表示
- **編集モード選択**: ポイント・ルート・スポット編集の切り替え
- **コントロールパネル**: 各モード専用の操作ボタン群
- **動的ポップアップUI**: ポイント・スポット位置に表示される入力ボックス
- **ステータス表示**: ポイント数・中間点数・スポット数のリアルタイム表示

マーカーの色分け・視覚表示 (更新)

- **赤い円**: 配置したポイント (IDラベル付き)
- **青い四角形 (■)** : 配置したスポット (名前ラベル付き)
- **橙色の菱形**: ルートの中間点
- **ルートライン**: 開始→中間点→終了を結ぶ連続線

**編集モード別の表示制御 :**

- **ポイント編集**: ポイントIDポップアップ表示、スポット入力ボックス非表示
- **ルート編集**: ポイントIDポップアップ表示 (灰色・強調) 、スポット入力ボックス非表示
- **スポット編集**: ポイントIDポップアップ非表示、スポット入力ボックス表示

ボタンの色分け統一

**全編集モード共通 :**

- **クリア系**: 赤色 (危険操作を表現)
- **JSON出力系**: 緑色 (成功・完了を表現)
- **補正系**: 黄色 (注意喚起)

この統一された色分けにより、操作の種類が直感的に理解できます。

## レスポンシブ対応

- **デスクトップ**: サイドバーレイアウト推奨
- **モバイル (768px以下)**: レイアウトの自動調整
- **ウィンドウリサイズ**: ResizeHandlerクラスによる最適化されたリサイズ処理

## データ構造

### ポイントJSON形式

```
{  
    "totalPoints": 3,  
    "imageReference": "sample.png",  
    "imageInfo": {  
        "width": 1920,  
        "height": 1080  
    },  
    "points": [  
        {  
            "index": 1,  
            "id": "A-01",  
            "imageX": 245,  
            "imageY": 387,  
            "isMarker": false  
        }  
    ],  
    "exportedAt": "2025-09-14T10:30:00.000Z"  
}
```

### ルートJSON形式

```
{  
    "routeInfo": {  
        "startPoint": "A-01",  
        "endPoint": "B-03",  
        "waypointCount": 5  
    },  
    "imageReference": "sample.png",  
    "imageInfo": {  
        "width": 1920,  
        "height": 1080  
    },  
    "points": [  
        {  
            "id": "A-01",  
            "x": 245, "y": 387,  
            "order": 1  
        },  
        {  
            "id": "B-03",  
            "x": 750, "y": 850,  
            "order": 2  
        }  
    ]  
}
```

```
{
    "type": "waypoint",
    "index": 1,
    "imageX": 320,
    "imageY": 450
}
],
"exportedAt": "2025-09-14T10:45:00.000Z"
}
```

## スポットJSON形式（更新）

```
{
    "totalSpots": 2,
    "imageReference": "sample.png",
    "imageInfo": {
        "width": 1920,
        "height": 1080
    },
    "spots": [
        {
            "index": 1,
            "name": "展望台",
            "imageX": 580,
            "imageY": 320
        }
    ],
    "exportedAt": "2025-09-14T11:00:00.000Z"
}
```

## 活用例

### 総合的な登山計画の作成

1. **ポイント編集**：主要ポイント（山頂、山小屋、分岐点等）をX-nn形式で体系管理
2. **スポット編集**：観光スポット（展望台、滝、神社等）を名前で直感管理
3. **ルート編集**：計画ルートを中間点で詳細設定・統合バリデーション確認
4. **統合管理**：3種類のデータを組み合わせて包括的な情報管理

### 実歩行記録の詳細記録

1. **ポイント記録**：ナビゲーション上重要なポイントを体系的に記録
2. **スポット記録**：実際に立ち寄った観光地・印象的な場所を名前で記録
3. **ルート記録**：実歩行ルートを中間点で精密再現
4. **精密調整**：改良されたドラッグ&ドロップ機能で位置を精密調整

### 地域情報の体系的管理

1. **ポイント管理**：管理対象エリアの重要な地点をID体系で管理

2. **スポット管理**：観光資源・特徴的な場所を名前で直感的に管理
3. **ルート管理**：推奨ルート・避難ルートを詳細設定
4. **品質管理**：統合バリデーション機能による一貫性確保

## チームでの情報共有

1. **標準化**：統一されたフォーマットによるデータ管理
2. **データ交換**：JSON形式によるチーム間データ共有
3. **役割分担**：
  - ポイント：ナビゲーション担当者
  - スポット：観光情報担当者
  - ルート：ルート計画担当者
4. **統合利用**：各担当が作成したデータを統合活用

## よくある質問・トラブルシューティング

Q: CORSエラーが発生して動作しません

A: **最重要**：必ずローカルサーバー経由でアクセスしてください。

```
python -m http.server 8000
# ブラウザで http://localhost:8000 を開く
```

ファイルを直接開く（file://）は動作しません。

Q: favicon.icoの404エラーが表示されます

A: v5.0で解決済みです：

- ♀ ピンアイコンが自動的に設定されます
- 404エラーは発生しなくなりました
- ブラウザタブでPointMarkerが識別しやすくなりました

Q: スポット編集時にポイントIDポップアップが表示されます

A: 最新版では仕様通り動作します：

- **スポット編集モード**: ポイントIDポップアップは完全に非表示
- **ポイント編集・ルート編集モード**: ポイントIDポップアップが表示
- モード切り替えで表示制御が自動実行されます

Q: ルート編集で開始・終了ポイントにエラーが出ます

A: 統合バリデーション機能による検証結果です：

- **存在エラー**: 指定したポイントIDが存在しない → 先にポイント編集でポイントを作成
- **重複エラー**: 開始・終了ポイントが同じ → 異なるポイントIDを指定
- **形式エラー**: X-nn形式でない → 正しい形式で入力（例：A-01）

Q: ポイントID名が重複してエラーが出ます

A: JSON出力時に統合検証を実行します：

- エラーメッセージで重複するIDを確認
- 該当するポイントのIDを修正してから再度出力
- 「ポイントID名の補正」で一括整理も可能

Q: ドラッグ&ドロップが滑らかでない

A: v5.0で大幅に改善されました：

- **統合ドラッグ処理**: DragDropHandlerクラスによる最適化
- **オブジェクト検出改善**: より正確で高速な検出
- **滑らかな移動**: フレームレート向上による快適な操作

Q: ポイントを移動できません

A: 編集モードを確認してください：

- **ポイント移動**: ポイント編集モードでのみ可能
- **スポット移動**: スポット編集モードでのみ可能
- **ルート編集モード**: ポイント・スポット移動は制限されます

Q: スポットマーカーが星形ではなく四角形です

A: v5.0で仕様変更されました：

- **新形状**: 四角形 (■) 青色
- **旧形状**: 星形 (★) 黄色
- **理由**: ポイントとの視認性向上、描画パフォーマンス改善

Q: 動作が重く感じます

A: v5.0でパフォーマンスが大幅改善されました：

- **コード効率化**: 重複処理の削除、統合クラス化
- **描画最適化**: 必要時のみの再描画
- **メモリ管理改善**: 効率的なオブジェクト管理

Q: JSONファイルが3種類あって混乱します

A: ファイル名で区別してください：

- **\*\_points.json**: ポイントデータ (X-nn形式ID管理)
- **\*\_spots.json**: スポットデータ (名前管理)
- **\*\_route\_\*.json**: ルートデータ (開始\_to\_終了形式)
- 各データは独立して管理・読み込み可能

## パフォーマンスガイド (改良版)

### 推奨使用環境

- **ポイント数:** 200個以下で最適性能（旧版：100個）
- **スポット数:** 100個以下で最適性能（旧版：50個）
- **ルート中間点:** 500個以下で最適性能（旧版：200個）
- **画像サイズ:** 4000x4000ピクセル以下を推奨（旧版：2000x2000）
- **ブラウザ:** Chrome/Edge最新版で最高性能

## 大量データ使用時

- **混在使用:** ポイント・スポット・ルートを同時大量使用時でも安定動作
- **メモリ管理改善:** より効率的なメモリ使用により安定性向上
- **定期保存:** 作業中は各機能で定期的にJSON保存（推奨）

## パフォーマンス改善内容

- **統合処理:** 類似機能の統合によりCPU負荷軽減
- **効率化:** リファクタリングによる処理速度向上
- **安定性:** エラーハンドリング改善による安定動作

## データバックアップ・管理

### 推奨バックアップ手順

1. **機能別保存:** ポイント・スポット・ルートを個別にJSON保存
2. **統合管理:** 同一地図の全データを同一フォルダで管理
3. **バージョン管理:** 作業節目での複数バージョン保存
4. **自動化:** 統合ファイル処理により効率的なバックアップ

### ファイル管理のコツ

- **命名規則:**
  - {地図名}\_points\_{作成日}.json
  - {地図名}\_spots\_{作成日}.json
  - {地図名}\_route\_{開始}to{終了}\_{作成日}.json
- **フォルダ整理:** 地域別・目的別でのフォルダ分類
- **セット管理:** 関連する3種類のJSONファイルをセットで管理

---

## プライバシーとセキュリティ

### データの安全性

- **完全ローカル処理:** すべての処理はブラウザ内で完結
- **外部送信なし:** インターネット接続や外部サーバーへのデータ送信は一切なし
- **ローカル保存のみ:** データはローカルファイルのみに保存
- **個人情報保護:** 個人情報の収集や追跡は一切行いません

### セキュリティ対策

- **ファイル形式制限:** PNG・JSON形式のみ受け入れ、不正ファイル拒否
- **統合検証:** Validatorsクラスによる一貫したセキュリティ対策

- **XSS対策:** 動的コンテンツの適切なエスケープ処理
- **入力検証:** 厳密なバリデーションによる不正データ排除

## 困ったときは

基本的なトラブル解決

1. **ブラウザ更新:** Ctrl+F5またはCmd+Rでページ更新
2. **ローカルサーバー確認:** <http://localhost:8000> でアクセスしているか確認
3. **ブラウザ変更:** Chrome/Edgeへの切り替えを推奨
4. **キャッシュクリア:** ブラウザキャッシュをクリアして再読み込み

## 詳細情報

- 操作方法がわからない場合は、この手引を再度ご確認ください
- 機能の詳細仕様については「funcspec.md」をご参照ください
- 開発者向け情報については「CLAUDE.md」をご参照ください

---

**最終更新:** 2025年9月14日 **バージョン:** 5.0 **更新内容:**

- リファクタリング後の改善内容を全面的に反映
- 統合オブジェクト検出システム・統一バリデーション機能の説明追加
- スポットマーカーの四角形（青色）への変更説明
- Favicon対応（♀ ピンアイコン、404エラー解消）の追加
- パフォーマンス改善内容の詳細説明（推奨環境の向上）
- 統合ドラッグ&ドロップ処理による操作性向上の説明
- 統一スタイル・エラー表示機能による一貫性向上の説明
- FAQ・トラブルシューティングの最新化（v5.0対応）
- データバックアップ・管理項目の充実化