

Leaflet と Cesium を切り替えて使用するライブラリ S-map の公開

西岡芳晴

産業技術総合研究所 地質情報研究部門
シームレス地質情報研究グループ



20万分の1日本シームレス地質図

～誰にでも使いやすい地質図を目指して～

シームレス地質図 3D



オフライン版



WMTS/WMS 配信



DVD 出版



メインビューア



<https://gbank.gsj.jp/seamless/>



S-map 公開サイト (試験公開中)

1. <https://gsj-seamless.jp/labs/smap/doc/>

Home
クイックスタート
API
解説
拡張ガイド
更新履歴

S-map

S-mapは、2015年5月に公開された日本シームレス地質図新ビューアのために作成されたJavaScript地図描画ライブラリです。

開発版	0.2.2	2016-08-09	↓ ダウンロード	http://gsj-seamless.jp/labs/smap/0.2.2/smap.js
最新安定版	0.2.1.1	2015-07-02	↓ ダウンロード	http://gsj-seamless.jp/labs/smap/smap.js
旧安定版	0.2.0.1	2015-07-02	↓ ダウンロード	

※ダウンロードファイルはsmap.js,ソースドキュメント,サンプルを含みます

※S-mapは、2016年8月9日現在試験公開中で、9月に正式公開を予定しています
 ※S-mapはApacheライセンスでの提供を予定しています。現段階でも利用は自由です
 ※このライブラリのサポートは計画しておりません

S-mapは、それ自体は地図描画を行う機能は持たず、Leaflet, Cesium等の地図描画ライブラリを呼び出して使用するための地図描画プラットフォームを提供します。このプラットフォームには、独自の地図描画エンジンを作成して組み込むことも可能です。

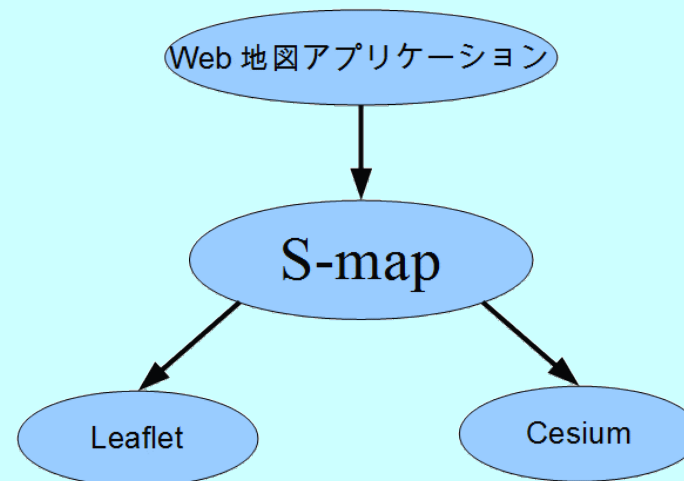
・特徴

- タイルレイヤーは動的にタイルを書き換え可能
- タイルレイヤーから画素情報, 凡例情報を取り出す機能を提供
- 地図を画像として出力する機能を提供
- KMLをサポート
- Promiseオブジェクトをサポートし, 非同期処理が記述しやすい
- 地図描画エンジン, コントロール, コントローラを拡張可能



“S-map” とは？

1. シームレス地質図のために開発した地図描画ライブラリ
2. 地図描画の中心部分はLeaflet, Cesiumが行っていて, それのラッパー.
3. S-map の S は
 - Seamless
 - Simple
 - Smart
 - Sophisticated



試験公開中

S-map の設計コンセプト

1. シンプル
 - ライブラリ利用者の負担を減らす
2. 高速
 - 地図描画エンジンの高速性を生かす
3. 賢明
 - かゆいところに手が届く

S-map の主な機能

1. 一般的な地図描画機能
 - タイルマップの描画
 - 複数レイヤー
 - マウス, タップによる地図移動, ズーム
 - 単独画像の描画
2. KML 対応
3. ゲームパッド対応
4. 地名検索対応

S-map – クイックスタート

- <https://gsj-seamless.jp/labs/smap/doc/quickstart.html>

Home
クイックスタート
API
解説
拡張ガイド
更新履歴

S-map – クイックスタート

準備

S-mapは単一のファイル `smap.js` で提供されます。 [ダウンロード](#)して適当なフォルダに配置して使うか、以下のURLを直接呼び出して利用してください。

`http://gsj-seamless.jp/labs/smap/smap.js`
 ※ただし、このURLは変更になる可能性があります。

また、S-mapはデフォルト設定では地図描画エンジンとして [Leaflet.js](http://leafletjs.com/) (<http://leafletjs.com/>) を利用します。Leaflet.jsもまた直接呼び出すか、ダウンロードして利用できるようにしてください。Leaflet.jsは専用のCSSも必要とするので、例えば以上を外部サーバ上から使って最小限のアプリケーションを作成すると、以下のようになります。

```

<!DOCTYPE>
<html height="100%">
<head>
  <meta charset="utf-8"/>
  <script src="http://leafletjs.com/dist/leaflet.js"></script>
  <link rel="stylesheet" href="http://leafletjs.com/dist/leaflet.css" />
  <script src="http://gsj-seamless.jp/labs/gsjmap/gsjmap.js"></script>
</head>
<body width="100%" height="100%" onload="Smap.map();" />
</html>
    
```

(サンプルファイル) [ミニマム版](#)



S-map シンプルアプリ

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="utf-8" />
  <style>
html, body { margin: 0px; width: 100%; height: 100% }
  </style>
  <script src="http://leafletjs.com/dist/leaflet.js"></script>
  <link rel="stylesheet" href="http://leafletjs.com/dist/leaflet.css" />
  <script src="https://gsj-seamless.jp/labs/smap/smap.js"></script>
</head>
<body>
  <script>
'use strict';
( function() {
  Smap.map();
} ) ();

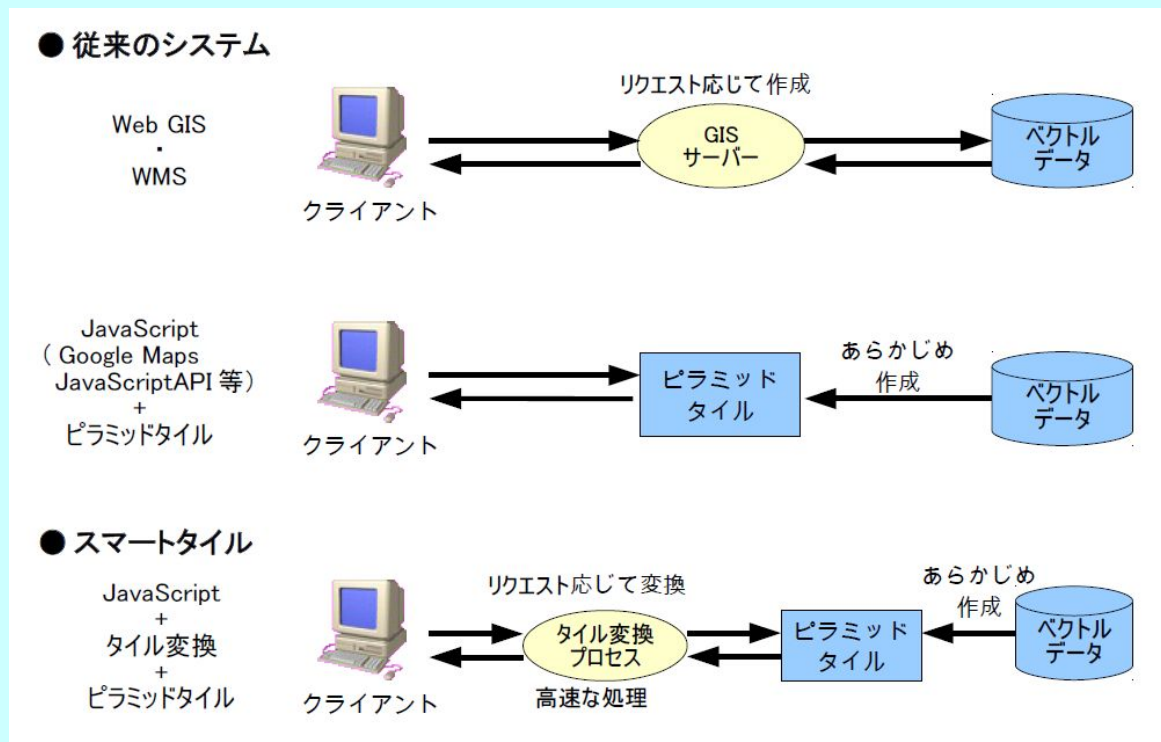
  </script>
</body>
</html>
```


S-map の特徴

1. シンプルインターフェイス
 - クラスを使わない関数ベース
2. スマートタイル支援
 - DrawTile, convertTile メソッド
3. ピクセル情報取得支援
 - getPixel, getPixelRGB, getPixelInfo メソッド
4. ダウンロード支援
 - CutImage, saveUrl, mapToKMZ メソッド
5. Promise のサポート
6. PNG 標高タイル対応

スマートタイル支援機能

1. ユーザの要望に応じて柔軟に表現を変更できる機能を提供
 - Layer.convertTile()
 - Layer.drawTile()

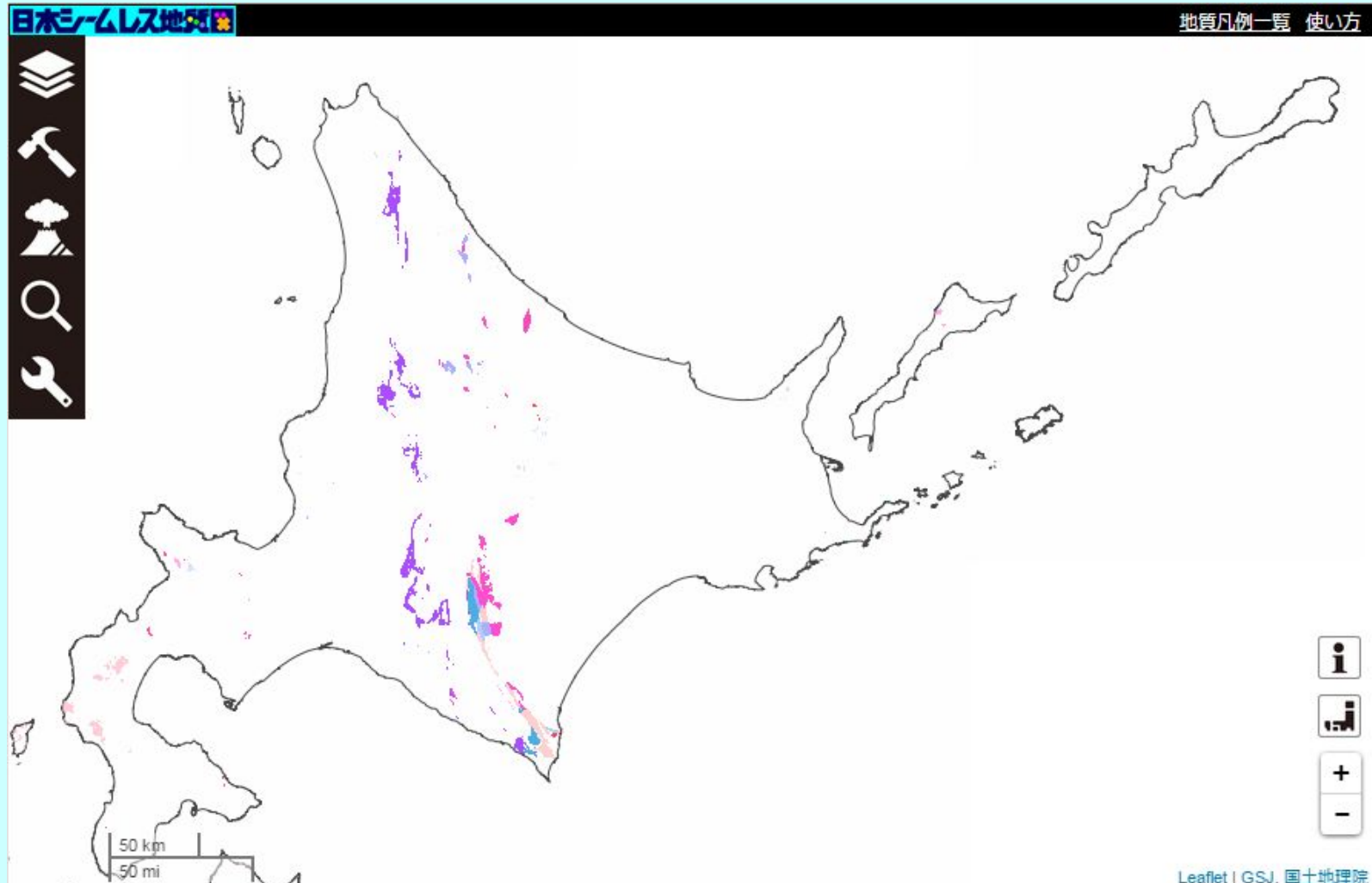


絞り込み機能



地質単元ごとの表示オン/オフ

北海道の深成岩だけ抽出（背景は白地図）



表示の On/Off の実装

```
"geologyOptions" : {  
  "url" : '../..../tilemap/{type}/{subtype}/{z}/{y}/{x}.png',  
  "title" : 'シームレス地質図',  
  "minZoom" : 5,  
  "maxZoom" : 13,  
  "convertTile" : function ( canvas_, coord ) {  
    var  
      ctx =canvas_.getContext( '2d' ),  
      imgData = ctx.getImageData( 0, 0, canvas_.width, canvas_.height ),  
      d = imgData.data;  
    if ( typeof app.legendColors !== 'undefined' ) {  
      for ( var i = 0, j = d.length; i < j; i += 4 ) {  
        var rgbInt = ( d[i+0] * 256 * 256 ) + ( d[i+1] * 256 ) + ( d[i+2] );  
        if ( typeof app.legendColors[ rgbInt ] !== 'undefined' ) {  
          d[ i + 3 ] = ( app.legendColors[ rgbInt ].checked ) ? 255 : 0;  
        }  
      }  
    }  
    ctx.putImageData( imgData, 0, 0 );  
  }  
},
```

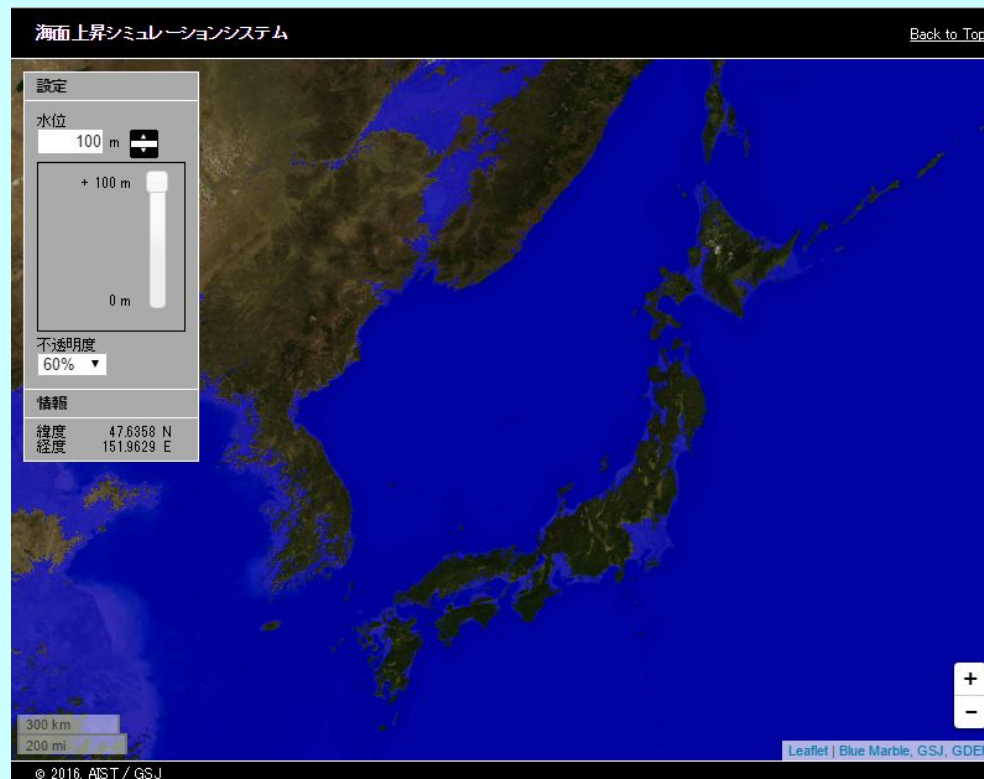
darwTile メソッドの利用

1. Layer オブジェクトの darwTile メソッドを記述して, PNG 標高タイルを元に計算し, タイルを描画
2. drawTile(canvas, coord)
 - canvas: タイルを描画する Canvas
 - coord: タイル座標
 - 戻り値: 処理が終了したときに成功となる Promise (非同期処理を含まないときは不要)

(利用事例)

海面上昇シミュレーションシステム

<https://gbank.gsj.jp/sealevel/>



地理院タイル (標高), Aster GDEM のデータを使用
drawTile を使って水没地域を青色に塗色

(利用事例)

傾斜量図テスター (試験公開中)

<http://gsj-seamless.jp/labs/elev/slope/slopetester.html>

傾斜量図テスター

背景

オーバーレイ1

不透明度

オーバーレイ2

不透明度

標高データ

標高の平滑化

計算式

傾斜量の平滑化

傾斜量を課税に変換する係数

傾斜量の増加に伴い、課税を

ガンマ補正

⌂ 暗くする、⌂ 明るくする

基準色

ダウンロード

※サイズが大きい等で失敗することがあります

このページでは、国土地理院の提供する標高タイルデータ及び基礎地図情報から得られた値(取組番号 平26情使 第284号)を使用しています。また、西之島(2014/12/4X)標高(国土地理院の「西之島付近噴火活動 標高タイル(2014年12月4日)」)を使用しています。Aster GDEMの標高データは経済産業省及びNASAIに帰属します。

ズームレベル: 12

N 35.3336 E 138.7731
H 1893.97m 傾斜角 30.9°

3 km
1 mi

Leaflet | 国土地理院, AIST GSI, シームレス地図図ラゴ

地理院タイル (標高) のデータを使用
drawTile を使って傾斜量を計算し、描画

ピクセル情報取得機能

1. 指定した位置のピクセルの情報を取得できる機能を提供
 - Layer.getPixel
 - 緯度経度とズームレベルを指定して, その位置の R, G, B 値を取得
 - Layer.getPixelRGB
 - 緯度経度とズームレベルを指定して, その位置の RGB 整数値 ($R*256*256 + G*256 + B$) を取得
 - Layer.getPixelInfo
 - 緯度経度とズームレベルを指定して, その位置の凡例情報を取得

凡例情報の表示

1. Layer オブジェクトの `getPixelInfo` メソッドを使用
2. `getPixelInfo(latLng, zoom, test)`
 - ... 前もって配列として指定した凡例情報から, ピクセルの色を基に凡例を抽出
 - `latLng`: 対象とする位置の緯度と経度
 - `zoom`: ズームレベル
 - `test`: 凡例の取得条件を変更するための関数 (省略可)
 - 戻り値: 検索終了時に成功する Promise (結果には該当した情報を含むオブジェクトが返される) .

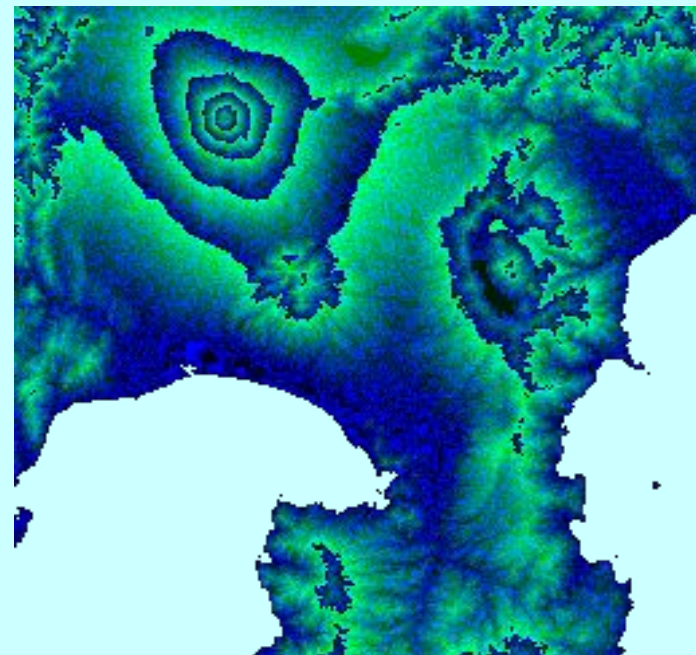


ダウンロード支援機能

1. 描画されている機能をクライアントに保存するための機能を提供
 - Map.cutImage(), Layer.cutImage()
 - 緯度経度で範囲を指定して画像を取得
 - Smap.saveURL()
 - urlで指定した内容をクライアントに保存
 - 画像, テキスト, dataURI等が指定可能
 - Smap.mapToKMZ()
 - Canvasと緯度経度範囲を指定して, Canvasに書かれた画像をKMZとして保存

PNG 標高タイル対応

1. Cesium 用 TerrainProvider を同梱
2. Cesium でのデフォルトの地形描画では, 地理タイル (標高) から作成した PNG 標高タイルを利用



PNG 標高タイルの例

PNG 標高タイルとは

1. PNG 標高タイルとは？

**Web ブラウザで利用するための
PNG を利用した標高タイルフォーマット**

2. 開発

- シームレス地質情報研究グループ

3. 目的

- 標高データの利用の高度化

4. 協力

- 国土地理院地理空間情報部

解説：西岡・長津 (2015b, 情報地質, vol. 27, p155-163)

仕様： <http://gsj-seamless.jp/labs/elev2/>



Promise のサポート

1. Promise は, 生成時点ではまだ完了していない処理を表現するオブジェクト
2. 非同期処理を簡潔に記述するために役立つ
3. S-map では, ブラウザがプロミスを実装している場合はそれを使用し, 実装していない場合は独自実装したものを使用

Promises サンプル

```
Smag.loadText( jsonsrc ).then(  
  function( text ) { // 成功時の処理  
    document.getElementById( 'out1' ).innerHTML =  
      '結果: ' + text.slice( 0, 40 );  
  },  
  function( error ) { // 失敗時の処理  
    document.getElementById( 'out1' ).innerHTML = error.msg;  
  }  
).then( function() {  
  // thenメソッドはさらに新しいPromiseを返すので  
  // メソッドチェーンでつなげられます。  
  document.getElementById( 'out2' ).innerHTML  
    = 'テキストファイル読み込み完了';  
} );
```

終わりに

1. Ver. 0.2.2 (dev)

- 現在最終テスト中
- ライセンス規定 (Apache License 2.0) 準備中
- 2016年12月 バージョン固定, 完成

2. Ver. 0.2.3

- Leflet 1.0.1 対応
- 2016年12月 公開

3. Ver. 1.0.0

- 正式版
- 2016年2月 公開予定

