

地理院地図の要素技術とその 利用について (B)

第二回

地理院地図パートナーネットワーク
会議

仕様に対する問題点

- 仕様の説明の充実
- Exampleの充実

仕様の説明の充実

- 仕様にある具体的な背景が欲しい
 - 仕様自体を作った開発の背景は何か
 - 何を解決するものか
 - 仕様を満たして何ができるのか

Exampleの充実

- ファイルの断片のようなものがあるが、見ただけではわかりづらい
 - レポジトリにexampleを置いて欲しい
 - JSのサンプル実装を作成して欲しい
- `maps.gsi.go.jp` で一つのファイルとして提供されている
 - 各機能をモジュール化して、一つのファイルにまとめるようにしてみてもどうか

geojson-with-style

- Styleは入れ子にしたほうがいいのでは
- Styleの使い回しをしたい

Styleの入れ子

```
{
  "type": "FeatureCollection", "features": [
  {
    "type": "Feature",
    "geometry": {"type": "Point", "coordinates": [135, 35]},
    "properties": {
      "名称": "〇〇公園",
      "住所": "〇〇県〇〇市〇〇",
      "_iconUrl": "http://cyberjapan.jp/symbols/010.png",
      "_iconSize": [20, 20],
      "_iconAnchor": [10, 10],
      "_className": "park"
    }
  }
  ]
}
```

アンダーバーが並んで
なんかいやだ

Styleの入れ子

```
{
  "type": "FeatureCollection", "features": [
  {
    "type": "Feature",
    "geometry": {"type": "Point", "coordinates": [135, 35]},
    "properties": {
      "名称": "〇〇公園",
      "住所": "〇〇県〇〇市〇〇",
      "style": {
        "iconUrl": "http://cyberjapan.jp/symbols/010.png",
        "iconSize": [20, 20],
        "iconAnchor": [10, 10],
        "className": "park"
      }
    }
  }
  ]
}
```

Style以下にいれた方が
すっきりするのでは

Styleの使い回し

```
{
  "type": "FeatureCollection", "features": [ {
    "type": "Feature",
    "geometry": {"type": "Point", "coordinates": [135, 35]},
    "properties": {
      "_iconUrl": "http://cyberjapan.jp/symbols/010.png"
    }
  },
  {
    "type": "Feature",
    "geometry": {"type": "Point", "coordinates": [135, 35]},
    "properties": {
      "_iconUrl": "http://cyberjapan.jp/symbols/010.png"
    }
  }
  ]
}
```

同じプロパティが連続するのは
データ作る側からすると面倒
ただし、_classNameでは解決
できない問題がある

Styleの使い回し

```
{
  "type": "FeatureCollection", "features": [ {
    "type": "Feature",
    "geometry": {...},
    "properties": {
      "_style": "symbols-010"
    }
  },
  {
    "type": "Feature",
    "geometry": {...},
    "properties": {
      "_style": "symbols-010"
    }
  }
]
}
```

```
{
  "symbols-010": {
    "_iconUrl": "http://cyberjapan.jp/symbols/010.png",
    "_iconSize": [20, 20],
    "_iconAnchor": [10, 10],
    "_className": "park"
  }
}
```

Styleのみを抽出して
JSON化するのはいかが
ファイルが二つになるので
ちょっと面倒だけど

直交座標系のサポート

- 直交座標系をサポートするようなGeoJSONが欲しい
 - propertiesにproj4のパラメータを追加するようなものでもよい
 - 地理院地図で重ねるときに標準でサポートされるとうれしい
 - ツール->作図ファイルからアップできるもので

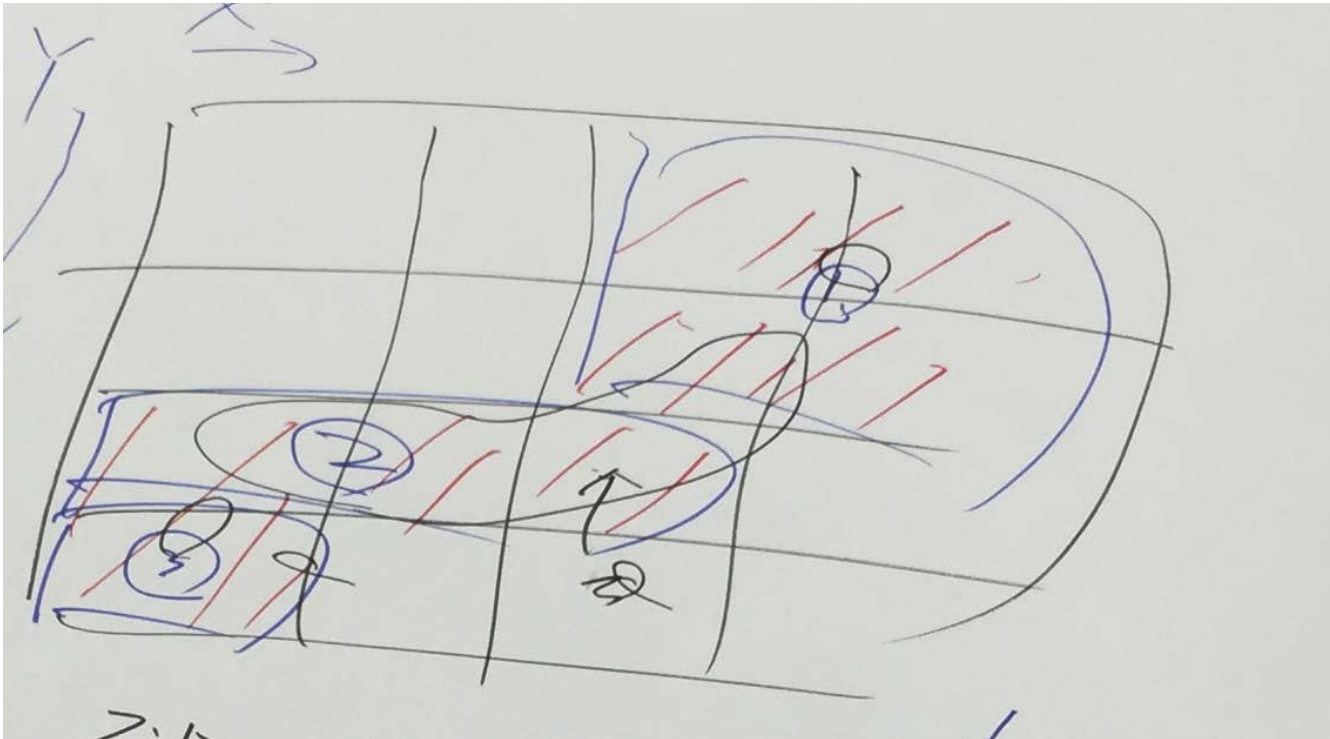
直交座標系API

- UTMGridのように直交座標系の系がわかるようなものが欲しい
- 座標ベースでAPIで叩いて取れるものでも良い
 - 系の境目がややこしい
 - 北海道、東京の島、島の半分で別れてるものとか

Cocotileの問題点

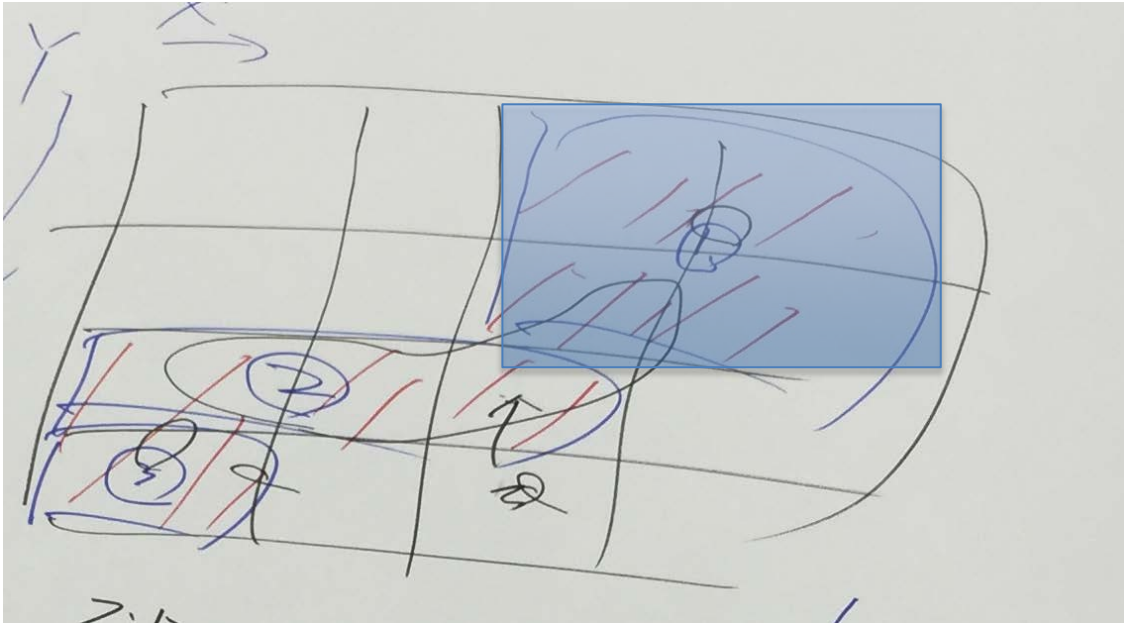
- Cocotileの仕様ではファイルがあるかどうかを先に調べるためにアクセスが必要となる
 - アクセス数が多い
- mokurokuはデカイ
 - 1.2GBとか辛い

アクセス



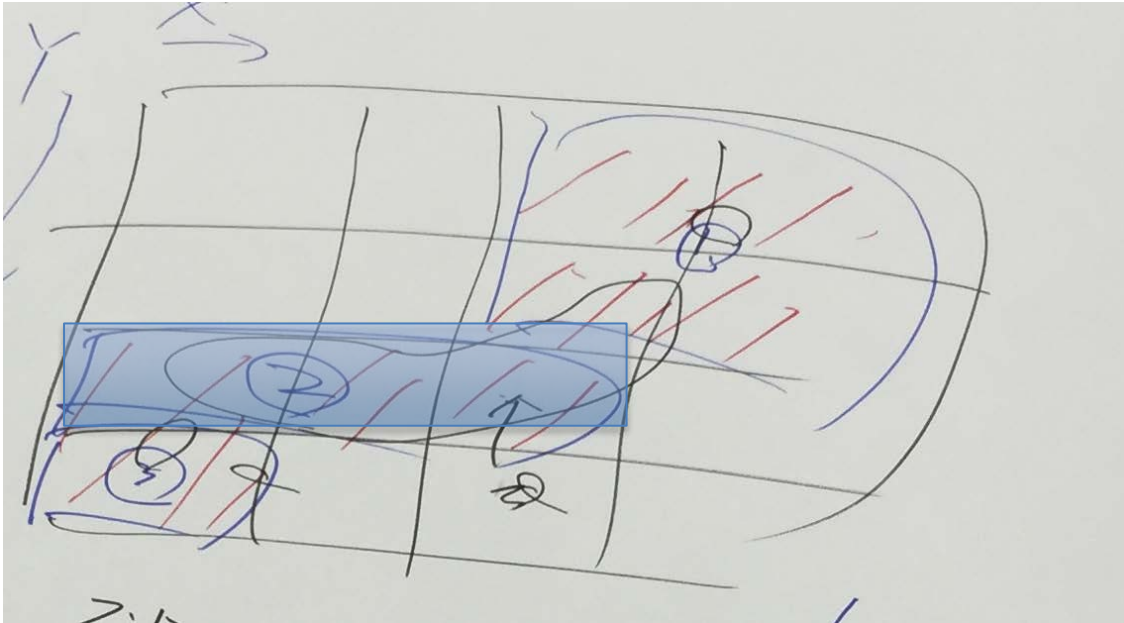
4x4=16アクセス必要
ズームレベルがあがると64アクセス
ネットワークアクセス自体は負荷となる

範囲のJSON



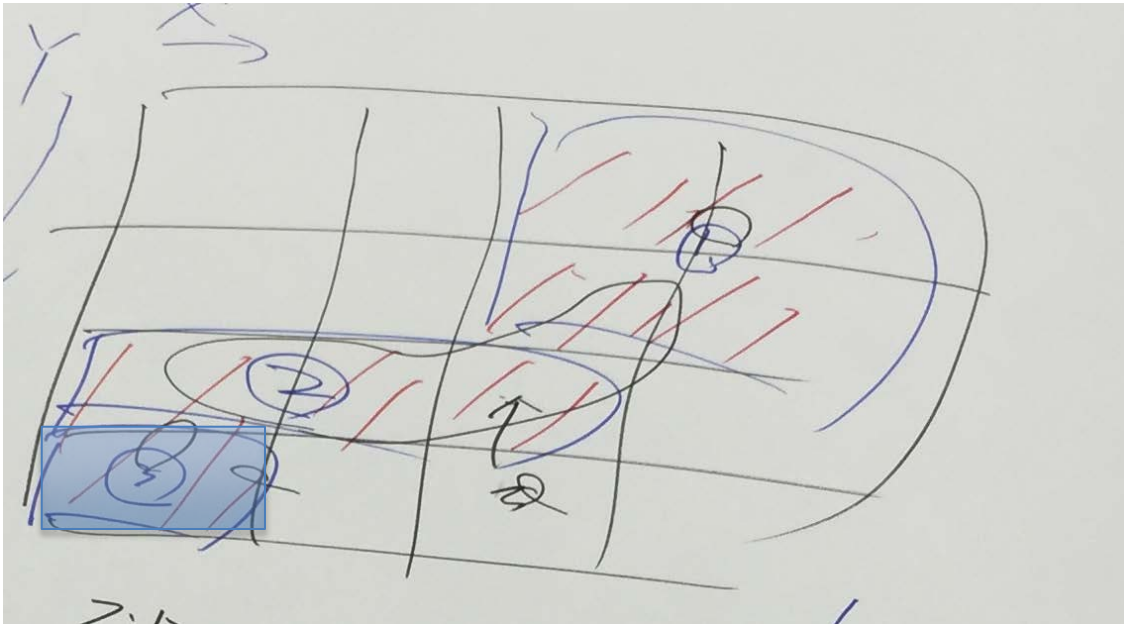
```
{
  startX: 2, startY:2,
  endX: 3, endY:3
}
```

範囲のJSON



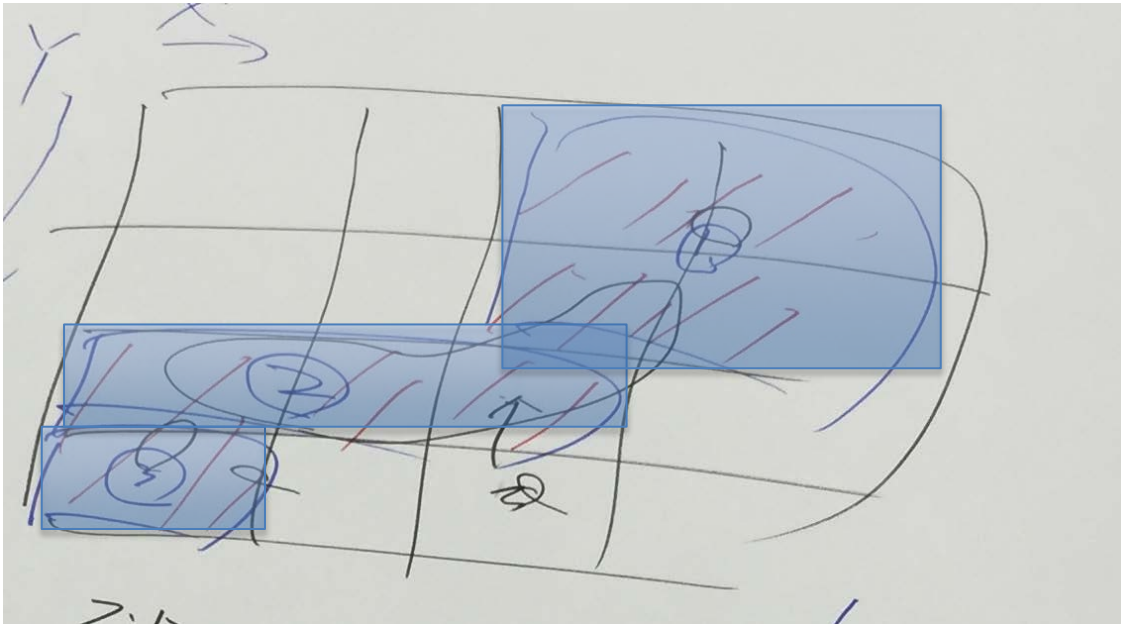
```
{  
  startX: 0, startY:2,  
  endX: 2, endY:2  
}
```

範囲のJSON



```
{  
  startX: 0, startY:3,  
  endX: 0, endY:3  
}
```


まとめる



```
[{
  startX: 2, startY:2,
  endX: 3, endY:3
},
{
  startX: 0, startY:2,
  endX: 2, endY:2
},
{
  startX: 0, startY:3,
  endX: 0, endY:3
}]
```

参考実装

- <http://cesiumjs.org/stk-terrain/tilesets/world/tiles/layer.json>
 - Cesiumの標高タイトルのLayerの範囲が取得できる
 - 1.2MBぐらいでzoom16までカバー
 - 日本の範囲ならそんなに大きくなるのでは？
 - 四角形を生成するアルゴリズムによる
 - layers-dot-txtを拡張するようなものでもいいかもしれない