

[hajime-miyauchi](#) / [webgis-hands-on](#)Branch: master ▾ [webgis-hands-on](#) / [mysql](#) / 01.md[Find file](#) [Copy path](#) hajime-miyauchi サンプルのURLを修正

852afdb 11 days ago

[1 contributor](#)

78 lines (59 sloc) 1.28 KB

MySQLから取得した緯度経度に円を描いてみよう

<http://play.mylab.jp/webgis-hands-on-mysql/mysql/sample-1-1/index.html>

index.htmlとmain.cssは変更なし

main.js

```
'strict'

var map = null;

function initMap() {
  map = new google.maps.Map($('#map').get(0), {
    center: {lat: 35.172899, lng: 136.887531},
    zoom: 15
  });

  $.ajax({
-   url: 'data.json',
+   url: 'api.php',
    type: 'GET',
    dataType: 'json'
  })
  .done(function(data){
    console.log(data);
    render(data);
  })
  .fail(function(){
    console.log('error');
  });
}

function render(data){
- data.circle_coordinates.forEach(function(coordinate){
+ data.forEach(function(geom){
  var circle = new google.maps.Circle({
    map: map,
-   center: coordinate,
+   center: {lat: geom.coordinates[1], lng: geom.coordinates[0]},
    radius: 100
  });
});
}
```

api.php

```
<?php

$db = mysqli_connect('localhost', <ユーザー名>, <パスワード>, 'opendata');

if(!$db){
  return;
}

$sql = <<<EOS
SELECT ST_AsGeoJson((geom)) AS geojson
FROM mcdnald
```

```
EOS;

$stmt = $db->prepare($sql);

$stmt->execute();
$stmt->bind_result($geojson);

$json = [];
while(($row = $stmt->fetch()) != NULL){
    $json[] = json_decode($geojson);
}

echo json_encode($json);

exit;
```

表示している範囲の緯度経度のみをMySQLから取得しよう

<http://play.mylab.jp/webgis-hands-on-mysql/mysql/sample-2-1/index.html>

index.htmlとmain.cssは変更なし

main.jsの変更点は下記2点。

- idle(待機)イベントが発生したら、idleMapsメソッドを呼ぶ
- api.phpに表示領域の東西南北の座標を緯度経度でパラメータとして渡す(図1)

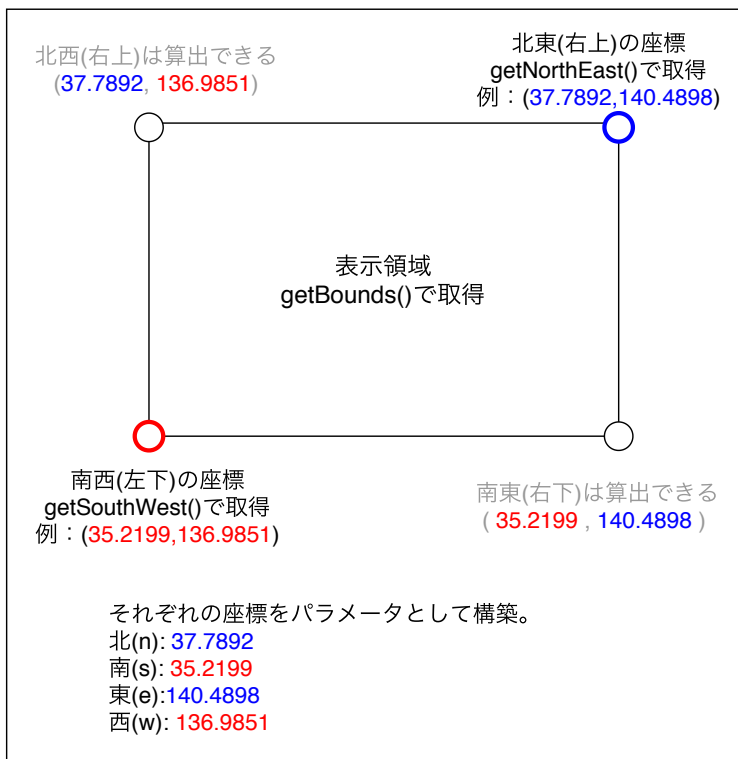


図1. 表示領域

api.phpの変更点は下記2点。

- パラメータとして渡された東西南北の緯度経度から、四角形のポリゴンデータを生成する(図2)
- 生成した四角形に含まれるデータのみを取得するようSQLに条件追加(図3)

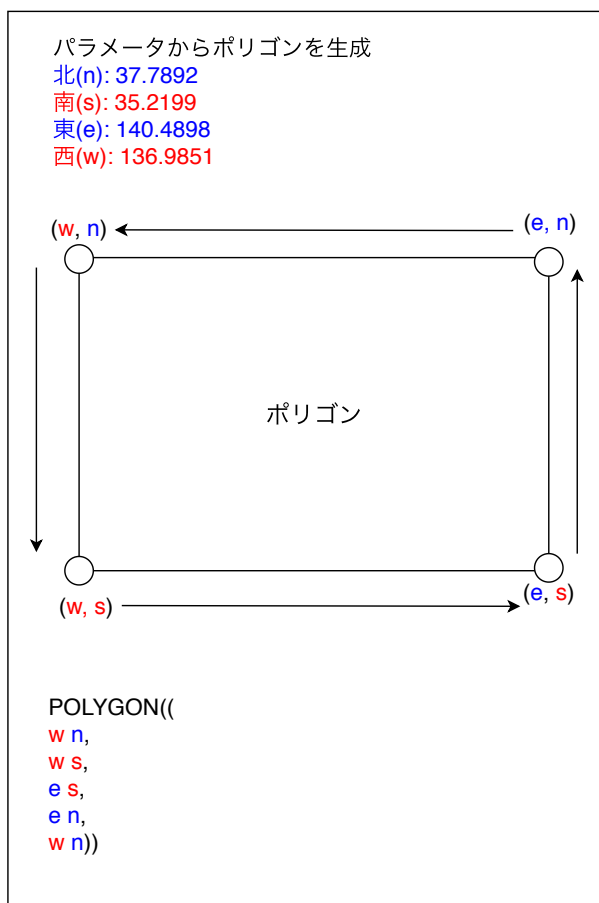


図2. ポリゴン

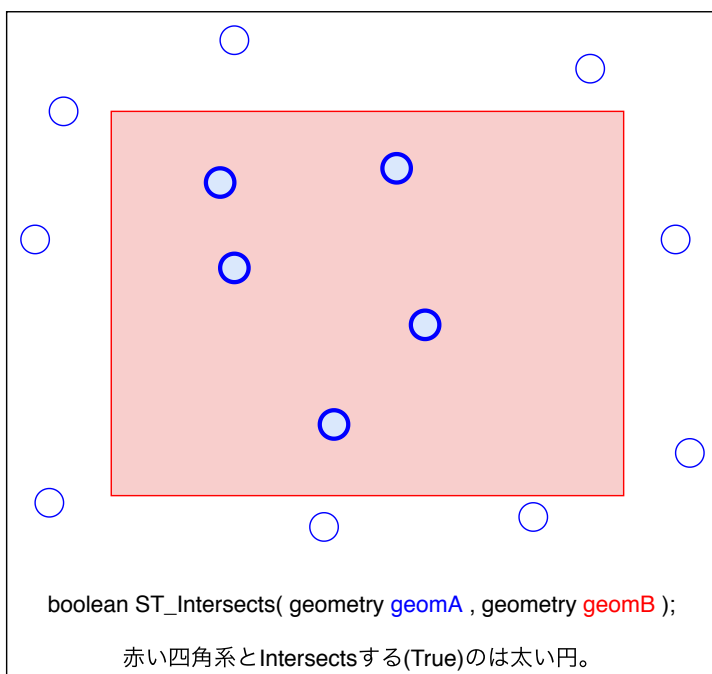


図3. ST_Intersects

main.js

```
'strict'

var map = null;

function initMap() {
  map = new google.maps.Map($('#map').get(0), {
    center: {lat: 35.172899, lng: 136.887531},
    zoom: 15
  });

  + google.maps.event.addListener(map, 'idle', idleMaps);
  +
  +
  +function idleMaps()
```

```

+{
+  var zoom = map.getZoom();
+  var bounds = map.getBounds();
+  var ne = bounds.getNorthEast();
+  var sw = bounds.getSouthWest();
+  var params = 'zoom=' + zoom + '&n=' + ne.lat() + '&s=' + sw.lat() + '&e=' + ne.lng() + '&w=' + sw.lng();
+
+  $.ajax({
-    url: 'api.php',
+    url: 'api.php?' + params,
    type: 'GET',
    dataType: 'json'
  })
  .done(function(data){
    console.log(data);
    render(data);
  })
  .fail(function(){
    console.log('error');
  });
}

function render(data){
  data.forEach(function(geom){
    var circle = new google.maps.Circle({
      map: map,
      center: {lat: geom.coordinates[1], lng: geom.coordinates[0]},
      radius: 100
    });
  });
}

```

api.php

```

<?php

$db = mysqli_connect('localhost', <ユーザー名>, <パスワード>, 'opendata');

if(!$db){
    return;
}

+$north = $_GET['n'];
+$west = $_GET['w'];
+$east = $_GET['e'];
+$south = $_GET['s'];
+$zoom = $_GET['zoom'];
+
+$geom = 'POLYGON((';
+$geom .= $west . ' ' . $north . ',';
+$geom .= $west . ' ' . $south . ',';
+$geom .= $east . ' ' . $south . ',';
+$geom .= $east . ' ' . $north . ',';
+$geom .= $west . ' ' . $north;
+$geom .= '))';
+
$sql = <<<EOS
SELECT ST_AsGeoJson((geom)) AS geojson
FROM mcdnald
+WHERE ST_Intersects(geom, ST_GeomFromText(?, 4326, 'axis-order=long-lat')) = True
EOS;

$stmt = $db->prepare($sql);
+$stmt->bind_param('s', $geom);
$stmt->execute();
$stmt->bind_result($geojson);

$json = [];
while(($row = $stmt->fetch()) != NULL){
    $json[] = json_decode($geojson);
}

echo json_encode($json);

```

上記のみでは再描画のたびに円が上書きされてしまう。そのため、再描画の際は描画済みの円を削除する必要がある。

<http://play.mylab.jp/webgis-hands-on-mysql/mysql/sample-2-2/index.html>

main.js

```
'strict'

var map = null;
+var objects = [];

function initMap() {
  map = new google.maps.Map($('#map').get(0), {
    center: {lat: 35.172899, lng: 136.887531},
    zoom: 15
  });

  google.maps.event.addListener(map, 'idle', idleMaps);
}

function idleMaps()
{
  var zoom = map.getZoom();
  var bounds = map.getBounds();
  var ne = bounds.getNorthEast();
  var sw = bounds.getSouthWest();
  var params = 'zoom=' + zoom + '&n=' + ne.lat() + '&s=' + sw.lat() + '&e=' + ne.lng() + '&w=' + sw.lng();

  $.ajax({
    url: 'api.php?' + params,
    type: 'GET',
    dataType: 'json'
  })
  .done(function(data){
    console.log(data);
    render(data);
  })
  .fail(function(){
    console.log('error');
  });
}

function render(data){
+  objects.forEach(function(o){
+    o.setMap(null);
+  });
+
  data.forEach(function(geom){
    var circle = new google.maps.Circle({
      map: map,
      center: {lat: geom.coordinates[1], lng: geom.coordinates[0]},
      radius: 100
    });
+
+    objects.push(circle);
  });
}
```