



実際にたたいてAPI

APIの初歩からプログラムまで使用方法のAtoZ

WebAPIとは？

インターネット(Web)を経由して情報や機能をやり取りする仕組み

パソコンやモバイルのアプリやWebサービスにより利用される



パソコン、スマホ

利用アプリ・サービス



統計



写真



地図



計算

なぜWebAPIなのか？



シンプル

特別な開発ツールや複雑なプログラミングが必要ない



多様性

パソコンやモバイル端末、AndroidやiOSなど様々な機種やOSから利用できる



マッシュアップ

複数のWebAPIを組み合わせて新たなサービスやソフトウェアを生み出せる

しずみちinfoのWebAPI

通信方式 : http(s)

データ形式 : GeoJSON

- 呼び出し形式

<https://openapi.city.shizuoka.jp/opendataapi/servicepoint/{サービスポイント名}>

- 道路規制情報のWebAPI

<https://openapi.city.shizuoka.jp/opendataapi/servicepoint/roadRegulation>

WebAPIの呼び出し

道路規制情報WebAPIを利用手順

① Wikiサイトのオープンデータ一覧でWebAPIのURLを確認する

Wikiサイト:

<<https://opendata-api-wiki-dot-shizuokashi-road.appspot.com>>

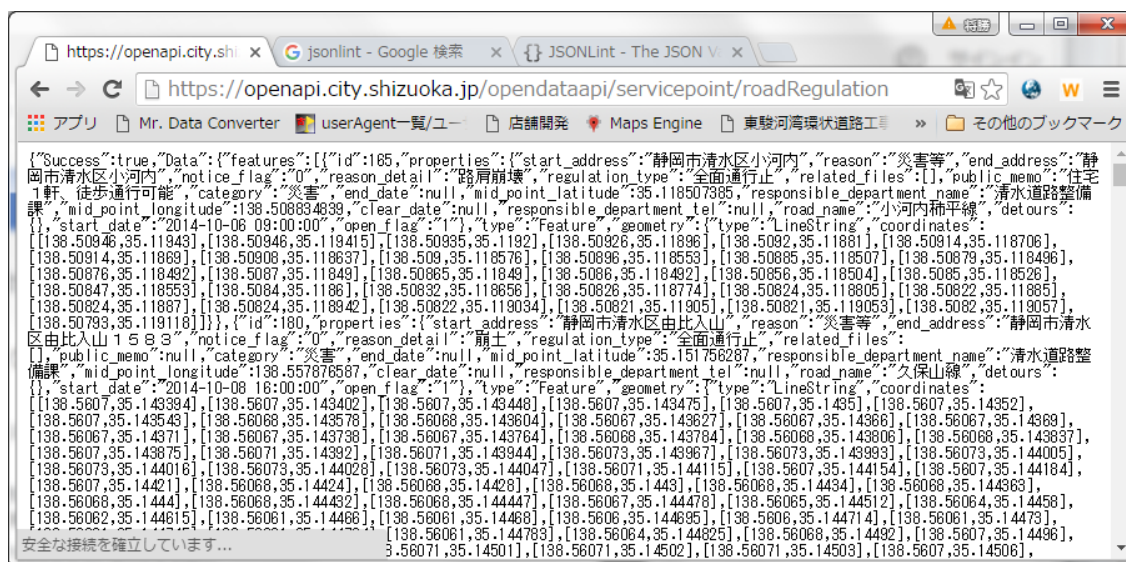
② Webブラウザを起動する

③ アドレスバー(URLバー)にWebAPIのURLを入力する

道路規制WebAPI:

<<https://openapi.city.shizuoka.jp/opendataapi/servicepoint/roadRegulation>>

●Webブラウザの路規制情報API 呼び出し結果



```
{
  "Success": true,
  "Data": {
    "features": [
      {
        "id": 185,
        "properties": {
          "start_address": "静岡市清水区小河内",
          "reason": "災害等",
          "end_address": "静岡市清水区小河内",
          "notice_flag": 0,
          "reason_detail": "路肩崩壊",
          "regulation_type": "全面通行止",
          "related_files": [],
          "public_memo": "住宅1軒、徒歩通行可能",
          "category": "災害",
          "end_date": null,
          "mid_point_latitude": 35.118507385,
          "responsible_department_name": "清水道路整備課",
          "mid_point_longitude": 138.508834839,
          "clear_date": null,
          "responsible_department_tel": null,
          "road_name": "小河内稲平線",
          "detours": [
            {
              "start_date": "2014-10-06 09:00:00",
              "open_flag": 1,
              "type": "Feature",
              "geometry": {
                "type": "LineString",
                "coordinates": [
                  [138.50946, 35.11943], [138.50946, 35.119415], [138.50935, 35.1192], [138.50926, 35.11896], [138.5092, 35.11881], [138.50914, 35.118706], [138.50914, 35.11869], [138.50908, 35.118637], [138.509, 35.118576], [138.50896, 35.118553], [138.50885, 35.118507], [138.50879, 35.118496], [138.50876, 35.118492], [138.5087, 35.11849], [138.50865, 35.11849], [138.5086, 35.118492], [138.50856, 35.118504], [138.5085, 35.118526], [138.50847, 35.118553], [138.5084, 35.1186], [138.50832, 35.118656], [138.50826, 35.118774], [138.50824, 35.118805], [138.50822, 35.11885], [138.50824, 35.11887], [138.50824, 35.118942], [138.50822, 35.119034], [138.50821, 35.11905], [138.50821, 35.119053], [138.5082, 35.119057], [138.50793, 35.119118]]],
              "id": 180,
              "properties": {
                "start_address": "静岡市清水区由比入山",
                "reason": "災害等",
                "end_address": "静岡市清水区由比入山 1583",
                "notice_flag": 0,
                "reason_detail": "崩土",
                "regulation_type": "全面通行止",
                "related_files": [
                  {
                    "public_memo": null,
                    "category": "災害",
                    "end_date": null,
                    "mid_point_latitude": 35.151758287,
                    "responsible_department_name": "清水道路整備課",
                    "mid_point_longitude": 138.557876587,
                    "clear_date": null,
                    "responsible_department_tel": null,
                    "road_name": "久保山線",
                    "detours": [
                      {
                        "start_date": "2014-10-08 16:00:00",
                        "open_flag": 1,
                        "type": "Feature",
                        "geometry": {
                          "type": "LineString",
                          "coordinates": [
                            [138.5607, 35.14394], [138.5607, 35.143402], [138.5607, 35.143448], [138.5607, 35.143475], [138.5607, 35.1435], [138.5607, 35.14352], [138.5607, 35.143543], [138.56068, 35.143578], [138.56068, 35.143604], [138.56067, 35.143627], [138.56067, 35.14366], [138.56067, 35.14368], [138.56067, 35.14371], [138.56067, 35.143738], [138.56067, 35.143764], [138.56068, 35.143784], [138.56068, 35.143806], [138.56068, 35.143837], [138.5607, 35.143875], [138.56071, 35.14392], [138.56071, 35.143944], [138.56073, 35.143967], [138.56073, 35.143993], [138.56073, 35.144005], [138.56073, 35.144016], [138.56073, 35.144028], [138.56073, 35.144047], [138.56071, 35.144115], [138.5607, 35.144154], [138.5607, 35.144184], [138.5607, 35.14421], [138.56068, 35.14424], [138.56068, 35.14428], [138.56068, 35.1443], [138.56068, 35.14434], [138.56068, 35.144363], [138.56068, 35.1444], [138.56068, 35.144432], [138.56068, 35.144447], [138.56067, 35.144478], [138.56065, 35.144512], [138.56064, 35.14458], [138.56062, 35.144615], [138.56061, 35.14466], [138.56061, 35.14468], [138.5606, 35.144695], [138.5606, 35.144714], [138.56061, 35.14473], [138.56061, 35.144783], [138.56064, 35.144825], [138.56068, 35.14492], [138.5607, 35.14496], [138.56071, 35.14501], [138.56071, 35.14502], [138.56071, 35.14503], [138.5607, 35.14506],
                          ]
                        }
                      ]
                    }
                  ]
                }
              }
            }
          ]
        }
      }
    ]
  }
}
```

GeoJSONとは？

JSONを拡張した地図やGISで利用することを想定したデータ
文字情報（属性情報）と位置情報（空間情報）をテキスト形式で保持
GoogleMapsやBingMapsなどの地図サービスが対応

● GeoJSONデータ

```
1 {  
2   "Success": true,  
3   "Data": {  
4     "type": "FeatureCollection",  
5     "features": [  
6       {  
7         "properties": {  
8           "observerpointname": "柚木",  
9           "waterlevel": -167,  
10          "roadname": "宮前大谷線",  
11          "observerpointcd": 50,  
12          "updatedate": "2016-08-19 14:50:00",  
13          "alertmode": "正常"  
14        },  
15        "type": "Feature",  
16        "geometry": {  
17          "type": "Point",  
18          "coordinates": [  
19            138.40733,  
20            34.980576  
21          ]  
22        }  
23      },  
24    ]  
  }  
}
```

文字情報

位置情報

● Google Mapsの表示例



なぜGeoJSONなのか？



シンプル

構造が単純でありプログラムによる利用や編集がしやすい。またXMLやCSVに比べてデータサイズが小さい。

わかりやすい

プログラムが容易に処理しやすく、人間が読むのもわかりやすい。

地図表示

Google MapsやBing Mapsなどの地図に表示できる。

JSONPとは？

Webアプリが外部のURL(ドメイン)からデータを取得する方法
JSONデータ(GeoJSONデータ)をJavaScriptの関数にしたデータ

しずみちinfoのWebAPIはJSONPを使用しないでデータ取得がおこなえる
IE8やIE9、Android4.2ブラウザなどレガシーなWebブラウザはJSONPを使用する

●JSONPデータ

```
callback( ← JavaScript関数
{ "Success": true,
  "Data": {
    "features": [
      "properties": {
        "observerpointname": " 柚木",
        "waterlevel": -177,
        "roadname": " 宮前大谷線",
        "observerpointcd": 50,
        "updatedate": "2016-08-30 14:15:00",
        "alertmode": " 通常"
      },
      "type": "Feature",
      "geometry": {
        "type": "Point", "coordinates": [138.40733, 34.980576]
      }
    ],
    "type": "FeatureCollection"
  }
});
```

●JSONデータ

```
{ "Success": true,
  "Data": {
    "features": [
      "properties": {
        "observerpointname": " 柚木",
        "waterlevel": -177,
        "roadname": " 宮前大谷線",
        "observerpointcd": 50,
        "updatedate": "2016-08-30 14:15:00",
        "alertmode": " 通常"
      },
      "type": "Feature",
      "geometry": {
        "type": "Point", "coordinates": [138.40733, 34.980576]
      }
    ],
    "type": "FeatureCollection"
  }
}
```

WebAPI利用アプリのプログラム

コードのサンプル

WebAPIをJavaScriptから呼び出すサンプル

<<https://storage.googleapis.com/shizuokashi-road.appspot.com/opendata-api-wiki/WebAPISample.html>>

● Wikiサイト



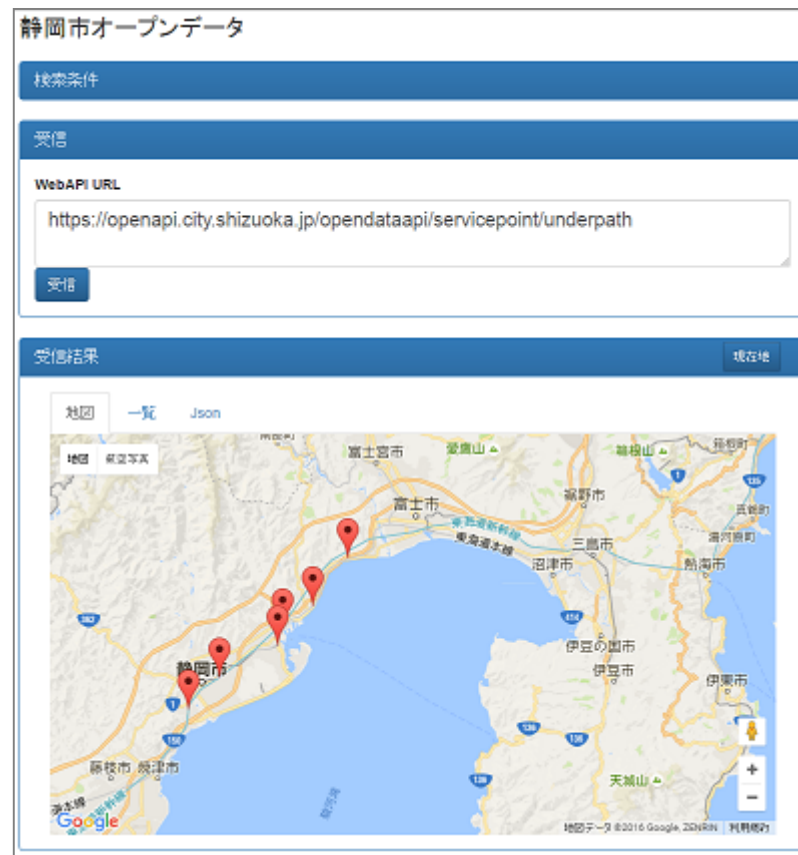
もっと!
しずみち info
クルマと道の未来を描く

メニュー

- トップページ
- オープンデータ一覧
- 利用ガイド
- APIリファレンス
- サンプルアプリ
- サンプルプログラム**
- 認証キー発行
- 利用規約



● サンプルプログラム



静岡市オープンデータ

検索条件


送信

WebAPI URL

送信

受信結果 現在地

地図 一覧 Json



地図 衛星写真

Google

地図データ ©2016 Google, ZENRIN | 利用規約

アプリからWebAPIを呼び出す

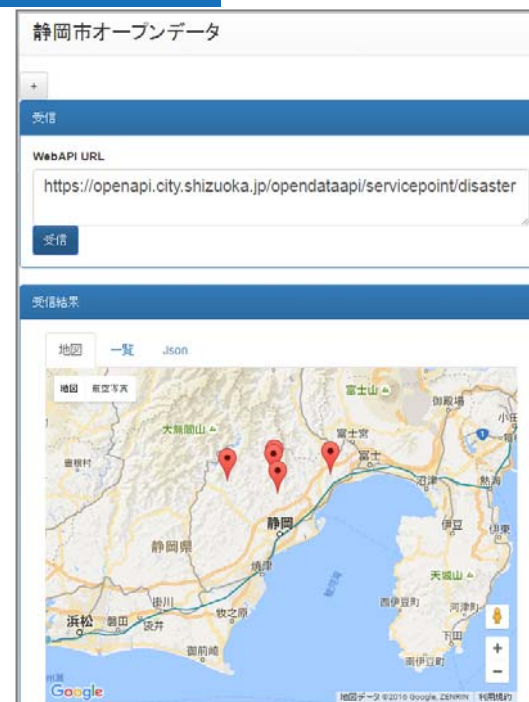
JavaScriptからオープンデータを呼び出し、地図、一覧に表示する

◎ サンプルコード

```
$("#getWebAPI").click(function() {  
    $.ajax({  
        type: "GET",  
        url: $("#textUrl").val(), //https://openapi.city.shizuoka.jp/opendataapi/servicepoint/disaster  
        contentType: 'application/json',  
        async: false,  
        success: function(json_data) {  
            addGeoJsonToGoogleMaps(googleMap, json_data.Data)  
            createTableFromGeoJSON("#dataTable", json_data.Data);  
            $("#dataJson").val(JSON.stringify(json_data, null, "  "));  
        },  
        error: function(e) {  
            window.alert('受信失敗!\n\n');  
            $("#memoJson").val("受信に失敗しました");  
        }  
    });  
});
```

//受信成功
//オープンデータを地図に表示する。
//オープンデータを一覧に表示する。
//オープンデータをJsonとして表示する

//受信失敗



アプリからWebAPIを呼び出す(JSONP)

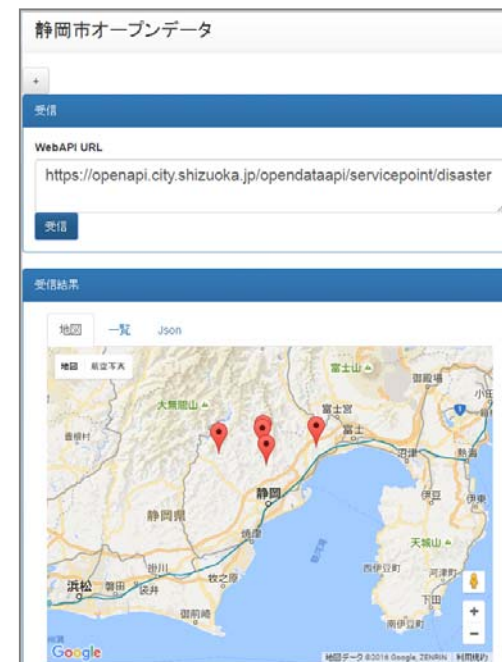
オープンデータをJavascriptの関数として取得

◎書式

/disaster?jsonp=true&callback={コールバック関数名}

◎呼び出し

/disaster?jsonp=true&callback=jsonpCallback



◎サンプルコード

```
$("#getJSONWebAPI").click(function() {  
    window.jsonpCallback = function(json_data) {  
        addGeoJsonToGoogleMaps(googleMap, json_data.Data);  
        addGeoJsonToTable("#dataTable", json_data.Data);  
        $("#dataJson").val(JSON.stringify(json_data, null, "  "));  
    };  
    var script = document.createElement('script');  
    var url = "https://openapi.city.shizuoka.jp/opendataapi/servicepoint/disaster?jsonp=true&callback=jsonpCallback";  
    script.setAttribute('src', url);  
    document.body.appendChild(script);  
});
```

//JSONPのコールバック関数を登録
//オープンデータを地図に表示する。
//オープンデータを一覧に表示する。
//オープンデータをJsonとして表示する

//javascriptオブジェクトを製紙絵する
//受信。BodyにWebAPIのURLを登録する。

空間検索

円形および矩形の領域指定によりデータを検索

◎検索パラメータ

lat:中心点の緯度

lng:中心点の経度

radius:中心点を起点とした半径
(メートル)

ある地点から半径4000メートル
にある道路規制データを検索

受信

WebAPI URL

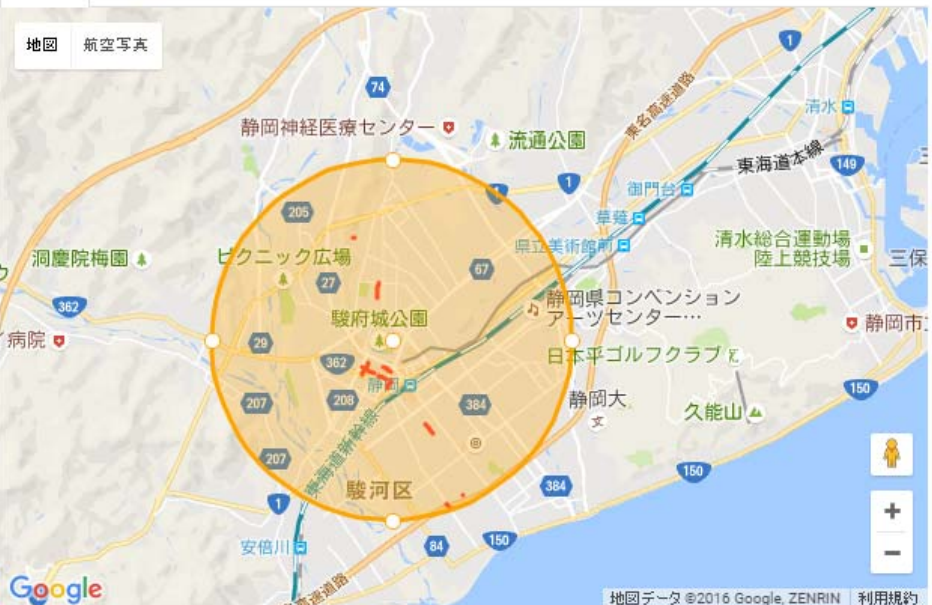
```
https://openapi.city.shizuoka.jp/opendataapi/servicepoint/roadRegulation?radius=4000&lat=34.98&lng=138.385
```

受信

受信結果

地図 一覧 Json

地図 航空写真



静岡神経医療センター 流通公園 駿府城公園 静岡大 久能山

Google 地図データ ©2016 Google, ZENRIN 利用規約

空間検索

◎ サンプルコード

```
$(function() {
  $("#textUrl").val("https://openapi.city.shizuoka.jp/opendataapi/servicepoint/disaster"); //WebURLの初期表示
  var googleMap = initGoogleMaps(); //Google地図の初期表示
  // 受信ボタン //////////////////////////////////////
  $("#getWebAPI").click(function() {
    $.ajax({
      type: "GET",
      url: "https://openapi.city.shizuoka.jp/opendataapi/servicepoint/roadRegulation?radius=4000&lat=34.98&lng=138.385",
      contentType: "application/json",
      async: false,
      success: function(json_data) { //受信成功
        addGeoJsonToGoogleMaps(googleMap, json_data.Data) //オープンデータを地図に表示する。
        createTableFromGeoJSON("#dataTable", json_data.Data); //オープンデータを一覧に表示する。
        $("#dataJson").val(JSON.stringify(json_data, null, " ")); //オープンデータをJsonとして表示する
      },
      error: function(e, dd, sss) { //受信失敗
        window.alert('受信失敗!\n\n');
        $("#memoJson").val("受信に失敗しました");
      }
    });
  });
});
```

属性検索

属性情報(データ内容)から指定した検索条件にあうデータを検索

規制区分:工事、開始日:9月1日以降の道路規制を検索

- ◎検索条件
- 完全一致
[フィールド名=検索値]
- 部分一致
[フィールド名=%検索値%]
- より大きい
[フィールド名=>検索値]
- より小さい
[フィールド名=<検索値]
- 一致しない
[フィールド名=!検索値]

受信

WebAPI URL


```
https://openapi.city.shizuoka.jp/opendataapi/servicepoint/roadRegulation?reason=工事&start_date=>2016-09-01 00:00:00
```

受信

受信結果

地図 一覧 Json

地図 航空写真



Google

地図データ ©2016 Google, ZENRIN 利用規約

属性検索

◎ サンプルコード

```
$("#getWebAPI").click(function() {  
  $.ajax({  
    type: "GET",  
    url: "https://openapi.city.shizuoka.jp/opendataapi/servicepoint/roadRegulation?reason=工事&start_date=>2016-09-01 00:00:00",  
    contentType: "application/json",  
    async: false,  
    success: function(json_data) {  
      //受信成功  
      addGeoJsonToGoogleMaps(googleMap, json_data.Data); //オープンデータを地図に表示する。  
      createTableFromGeoJSON("#dataList", json_data.Data); //オープンデータを一覧に表示する。  
      $("#dataJson").val(JSON.stringify(json_data, null, "  ")); //オープンデータをJsonとして表示する  
    },  
    error: function(e, dd, sss) {  
      //受信失敗  
      window.alert('受信失敗!\n\n');  
      $("#memoJson").val("受信に失敗しました");  
    }  
  });  
});
```


ページング

一回の呼び出しにおけるデータ件数に制限をかけている

初期設定：30件 最大設定：100件

大量のデータを取得する場合はページ単位に分割で読みこむ

◎ 検索条件

page={ページ番号}&row={データ件数}

◎ 道路規制を読み込む

1ページ目：/roadRegulation?page=1&row=30

2ページ目：/roadRegulation?page=2&row=30

データ1	1ページ
データ2	
データ3	
⋮	
データ28	
データ29	
データ30	
データ31	2ページ
データ32	
データ33	
⋮	
データ58	
データ59	
データ60	

並び順

データ取得時に路線名や日時などの属性項目で並び替える

並び順とページングを同時に指定することにより、ページ切り替え画面を作成する

◎ 検索条件

order={フィールド名}+{asc:昇順/desc:降順}

◎ 道路規制情報を規制開始日により並べる

昇順 :

/roadRegulation?order=start_date

降順 :

/roadRegulation?order=start_date+desc

◎ 並び順とページングを同時に指定する

/roadRegulation?order=start_date&page=2&row=30

● 昇順

2016/8/01 08:30
2016/8/08 08:00
2016/8/15 09:36
2016/8/16 21:00
2016/8/17 08:30

● 降順

2016/8/17 08:30
2016/8/16 21:00
2016/8/15 09:36
2016/8/08 08:00
2016/8/01 08:30

認証キー

通常利用では全利用者の利用回数が10秒あたり10回を超えると利用を制限している
認証キーを利用することにより、認証キーごとに10秒あたり10回まで利用できる

認証キーの発行サイト

<https://opendata-api-wiki-dot-shizuokashi-road.appspot.com/authkey/create>

● オープンデータWikiサイト



The screenshot shows the 'authkey' page on the Open Data Wiki site. The page title is 'オープンデータ 認証キー発行' (Open Data Authentication Key Issuance). It contains instructions: 'メールアドレスの入力をおこない、発行ボタンを押してください。メールアドレス宛てに、認証キーが記載されたメールを送信します。' (Enter the email address and click the issuance button. An email containing the authentication key will be sent to the email address.) Below the instructions is a text input field for the email address and a '発行' (Issue) button. The left sidebar has a menu with '認証キー発行' (Issue Authentication Key) highlighted in a red box.

● 認証キーの発行メール



The screenshot shows an email received from 'shizu.sns.noreply@gmail.com' with the subject '静岡市オープンデータAPI: 認証キー発行' (Shizuoka City Open Data API: Authentication Key Issuance). The email content is as follows:

この度は、静岡市オープンデータ WebAPI へ登録いただき有難うございます。

認証キーの発行が完了しました。
認証キー：
556d21c611zaa5f6bb30b5a529c26d35f72b7d630277zzaz5aaf5cad9526566d

ご利用方法は以下をご覧ください。
認証キーの利用方法：
<http://opendata-api-wiki-dot-shizuokashi-road.appspot.com/reference#parameters_authorization>
WebAPI の利用方法：
<<http://opendata-api-wiki-dot-shizuokashi-road.appspot.com/guide>>
オープンデータ一覧：
<<http://opendata-api-wiki-dot-shizuokashi-road.appspot.com/resource/list>>

本メールへの返信は受け付けておりませんのでご注意ください。
このメールに心あたりがない場合は、
<<http://opendata-api-wiki-dot-shizuokashi-road.appspot.com/authkey/>>

認証キーの利用

WebAPIにクエリパラメータとして設定する

◎ 設定方法

/roadRegulation?auth_key={認証キー}

◎ 呼び出し

/roadRegulation?auth_key=556d21c611zaa5f6bb30b5a529c26d35f72b7d630277z

受信


WebAPI URL

```
https://openapi.city.shizuoka.jp/opendataapi/servicepoint/disaster=auth_key&556d21c611zaa5f6bb30b5a529c26d35f72b7d630277z
```

受信

受信結果

地図 一覧 Json



The map displays the Shizuoka region, including parts of Aichi, Yamanashi, and Kanagawa Prefectures. Three red pins are placed on the map, indicating disaster service points. The map includes labels for various locations such as 富士山 (Mount Fuji), 静岡 (Shizuoka), and 浜松 (Hamamatsu). The map interface includes a search bar, a list view, and a JSON view option.

Copyright (C) 2016 ZENRIN CO.,LTD.(Z16KC第464号) ©Google

参考

静岡市オープンデータWiki

<http://opendata-api-wiki-dot-shizuokashi-road.appspot.com>

Google Map API V3 ガイド

<https://developers.google.com/maps/documentation/javascript/tutorial>

JavaScriptにおけるJsonデータの操作ガイド

https://developer.mozilla.org/ja/docs/Web/JavaScript/Reference/Global_Objects/JSON