

API

Contents

- APIとは
- google books API
 - APIの紹介
 - データ取得
- bing maps API
 - APIキーの取得
 - 現在地の取得
 - 地図表示
 - ピンを立てる&吹き出し表示
- 課題発表→P2P(演習)タイム

rules...

- 授業中は常にエディタを起動！
- 考えたことや感じたことはslackのガヤチャンネルでガンガン発信！
- 質問はslackへ！ 他の人の質問にも目を通そう！（同じ質問があるかも）
- 演習時，できた人はスクショなどslackに貼ってアウトプット！
- まずは打ち間違いを疑おう！
 - `{'";` など
- 書いたら保存しよう！（よく忘れる！）
 - `command + s`
 - `ctrl + s`

本日のゴール

- 外部サービスを使う！
- API使用の流れを把握する！
- できることの発想を広げる！

Applocation Programming Interface

What is API??

- APIはアプリケーションの一部を公開したもの.
- すごい人が作ったものを使わせてもらえるイメージ！
- 例
 - maps API : リクエストに応じてブラウザに地図を表示する
 - google books api : google booksのデータを返す
 - open weather map API : 天気情報を返す

What is API??

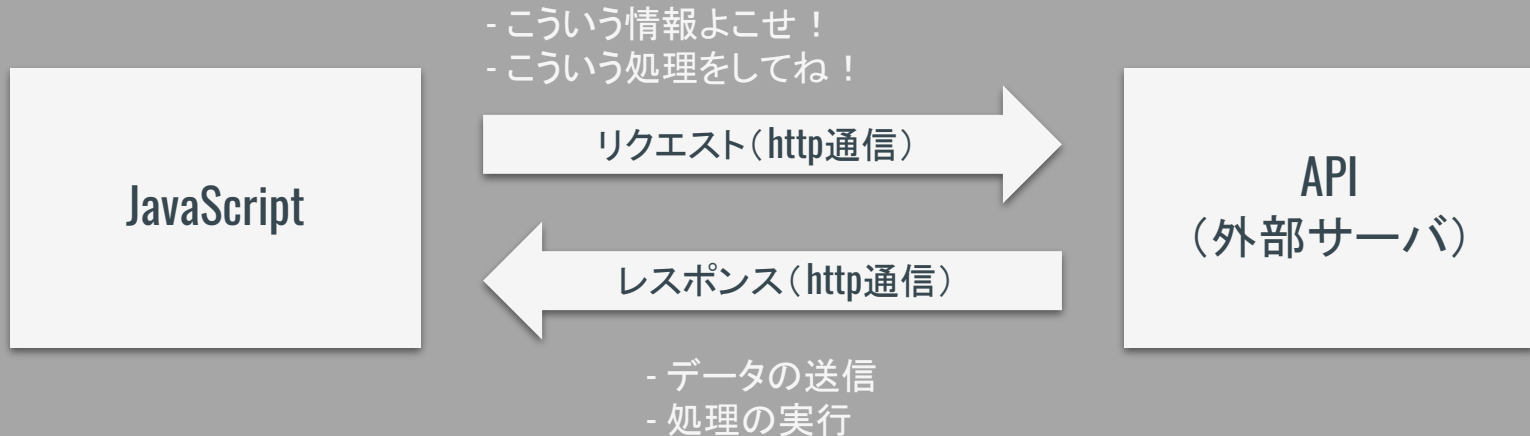
利点

- 他社のデータを利用できる！
- 自分で作らなくて良いので開発速度がアップ！
- いろいろなデータを組み合わせてアプリケーションを実装できる！

スタートアップや個人開発に最適！

基本的な仕組み

- 基本は「リクエスト」と「レスポンス」
- 「http通信」でデータをやり取りする



webアプリケーションの構造そのもの！！

タイプは大きく分けて2種類

- データをもらうタイプ(こちらが一般的)
 - 検索条件などを指定して, 条件に合う情報を送ってもらう.
 - google books API : 本のタイトルや著者で検索.
 - open weather mapAPI : 場所で天気を検索.
 - 画像認識 or 音声認識: 取得したデータを指定の形式で返す.
- 処理をしてもらうタイプ
 - 条件を指定して, 実際にブラウザ表示などをやってもらう.
 - bing maps API : 場所などを指定して地図を表示する.
 - google maps API : Google Map
 - (firebase)

Google books API

Google books API

- google books API
 - 今回は本を検索するAPIを使用しましょう！
 - ドキュメント:<https://developers.google.com/books/docs/overview>
- できること
 - google booksに登録されている本のjsonデータを取得できる！
 - タイトルや著者などで条件を指定できる！
 - （仕様をよく読みましょう！）
 - API keyが不要！！！！

【参考】APIキーとは

- APIキー
 - APIにリクエストを送るためにはユニークな文字列が必要.
 - 設定を行ったアプリケーションのみ動作させるため.
 - (無断で使用されないように. . . !)
 - この文字列をAPIキーと呼び, APIの種類問わず必須の場合が多い.

Google books API

- 準備
 - 下記のURLにリクエストしよう！
 - <https://www.googleapis.com/books/v1/volumes>
- パラメータの追加
 - 今回はタイトルに「firebase」が入っている本を指定して検索してみよう！
 - 上記URLの最後に「?q=intitle:firebase」を追加しよう！
- リクエストにはaxiosライブラリを使う！！
 - JavaScriptでhttp通信を行うためのライブラリ(読み込みが必要)
 - 最近のモダンなフレームワーク(react, vue.jsなど)でも利用されている

Google books API

```
// APIにリクエストを送るときはaxios.get()を使用
axios.get(url)
  .then(function (response) {
    // リクエスト成功時の処理 (responseに結果が入っている)
    console.log(response);
  }).catch(function (error) {
    // リクエスト失敗時の処理 (errorにエラー内容が入っている)
    console.log(error);
  }).finally(function () {
    // 成功失敗に関わらず必ず実行
    console.log('done!');
  });
```

※ console.log()以外は決まっている！！

実行結果

VM130:3

```
▼ {kind: "books#volumes", totalItems: 90, items: Array(10)}
```

```
▼ items: Array(10)
```

data.itemsにデータが入ってくる

```
▼ 0:
```

- ▶ accessInfo: {country: "JP", viewability: "PARTIAL", embeddable: true, publicDomain...
- ▶ etag: "byZj+et6VUU"
- ▶ id: "nbZ-DwAAQBAJ"
- ▶ kind: "books#volume"
- ▶ saleInfo: {country: "JP", saleability: "FOR_SALE", isEbook: true, listPrice: {...}}
- ▶ searchInfo: {textSnippet: "【FlutterとFirebaseでクロスプラットフォームアプリを作る！】 ...
selfLink: "https://www.googleapis.com/books/v1/volumes/nbZ-DwAAQBAJ"
- ▶ volumeInfo: {title: "Flutter×Firebaseで始めるモバイルアプリ開発", authors: Array(2), ...
- ▶ __proto__: Object

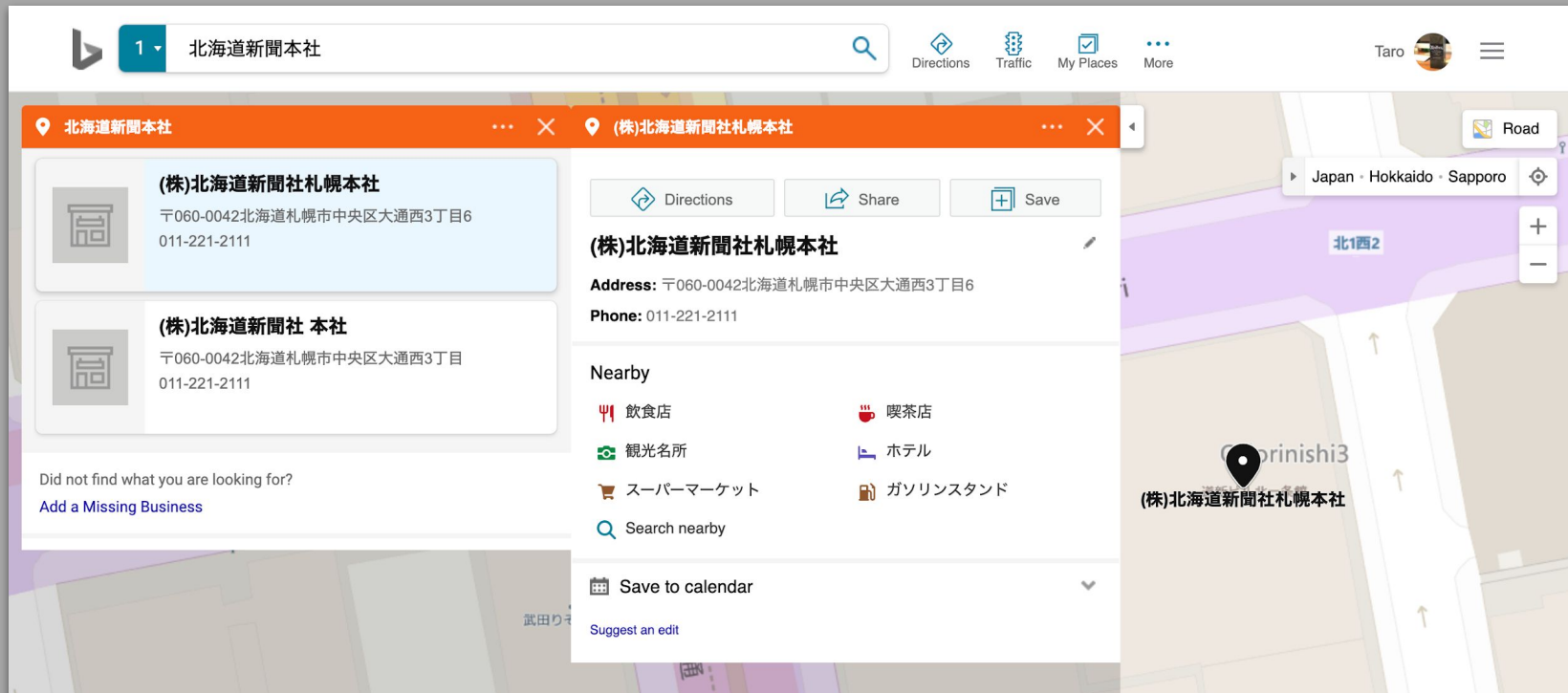
Google books APIの練習

google books APIを使って下記を実行しよう！

- パラメータを自由に設定してURLを作成しよう！
- ボタン押下時に、本の情報を`console.log()`で出力しよう！
- できる人はブラウザ上に表示もしてみよう！
- (タイトルを表示してリンクを張ってみよう！)

bing maps API

bing maps API



bing maps API

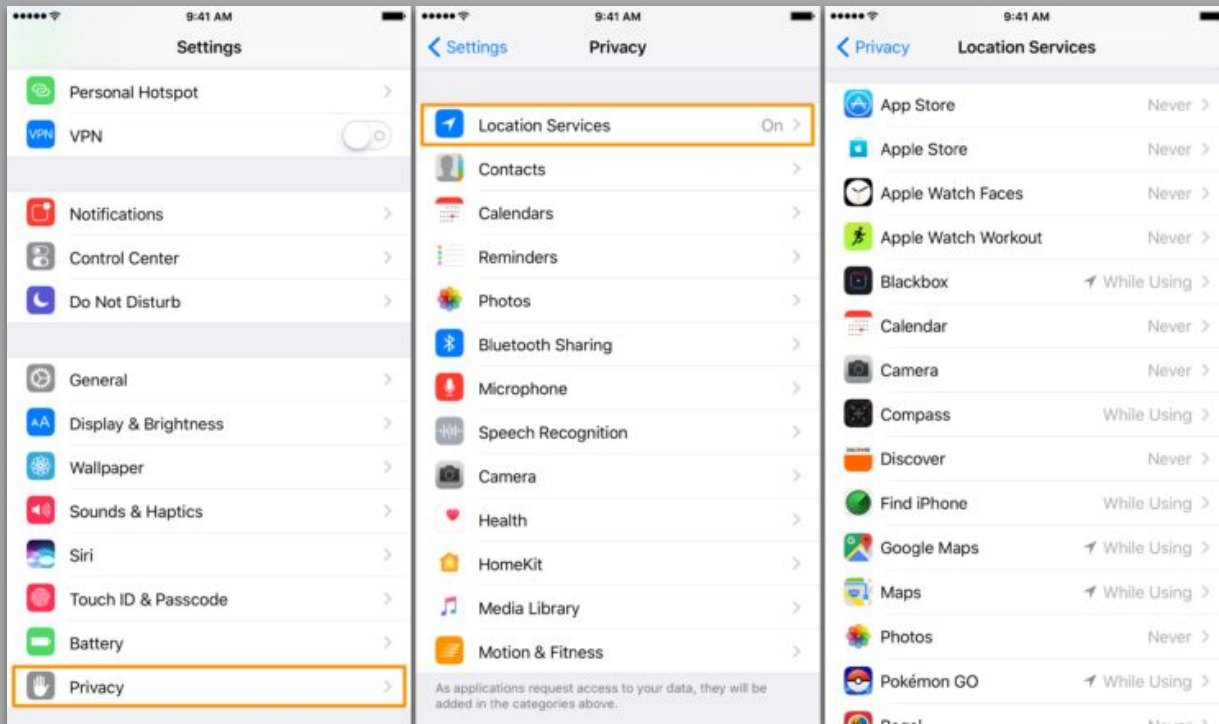
Microsoftが提供する地図API

- 位置情報を利用して様々な機能を提供する！
- 地図の表示, ルート検索
- ピンの設置, infobox, etc...
- 住所から緯度経度を取得
- 緯度経度から住所を取得
- 無料

位置情報

位置情報の利用

- 世の中のサービスの8割程度が利用..!!



位置情報の利用

地図 / ルート検索

- 位置情報とマップの組み合わせ
 - 今まではユーザの現在地がわからなかった
- 地図上に店舗データなどを表示
 - 住所だけではわかりにくい
- どこに行けばよいかがわかる
 - 道を間違っても自動的に現在地からのルートが分かる

位置情報の利用

店舗のマーケティングリサーチ

- 電波強度を測定
 - 強ければ店内
- 滞在時間で顧客を分ける
 - 長ければ店内, 短ければ店外
- 営業時間で調べる
 - 営業時間内なら顧客, 時間外は通行人

位置情報の利用

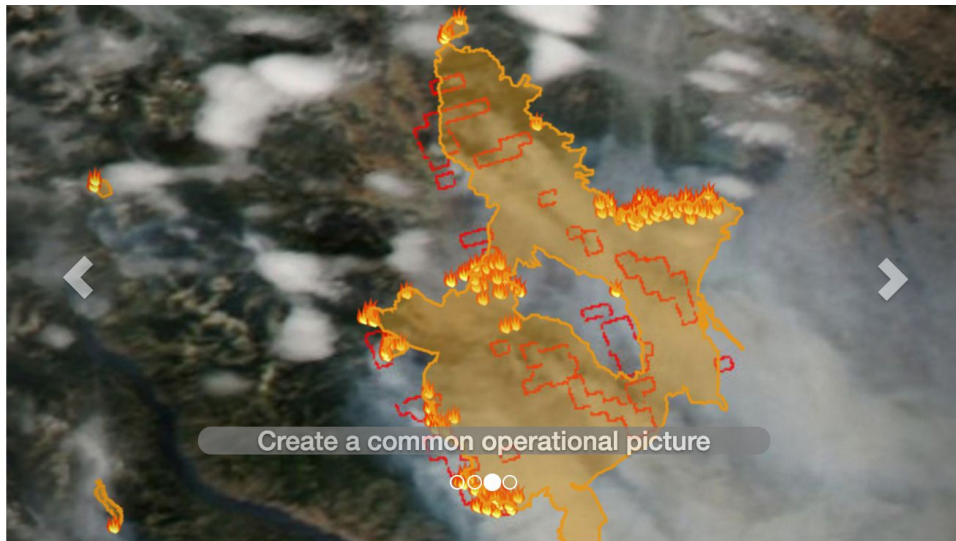
店内の販促ツール

- 棚の近くにビーコンを設置
- ユーザが近づくとLINEにクーポンを配信
 - ユーザにアプリ等のツールが必要ない
- 購入レシートを送信するとポイントが当たる
 - 再度の来店を促す

APIキーの取得

<https://www.bingmapsportal.com>にアクセス！

Bing maps | Dev Center



Welcome

The Bing Maps Dev Center provides the tools and resources you need to develop with Bing Maps. You can store, access, and keep track of your store locations or other spatial data through our online data source management system. You will also receive important announcements around your Bing Maps account.

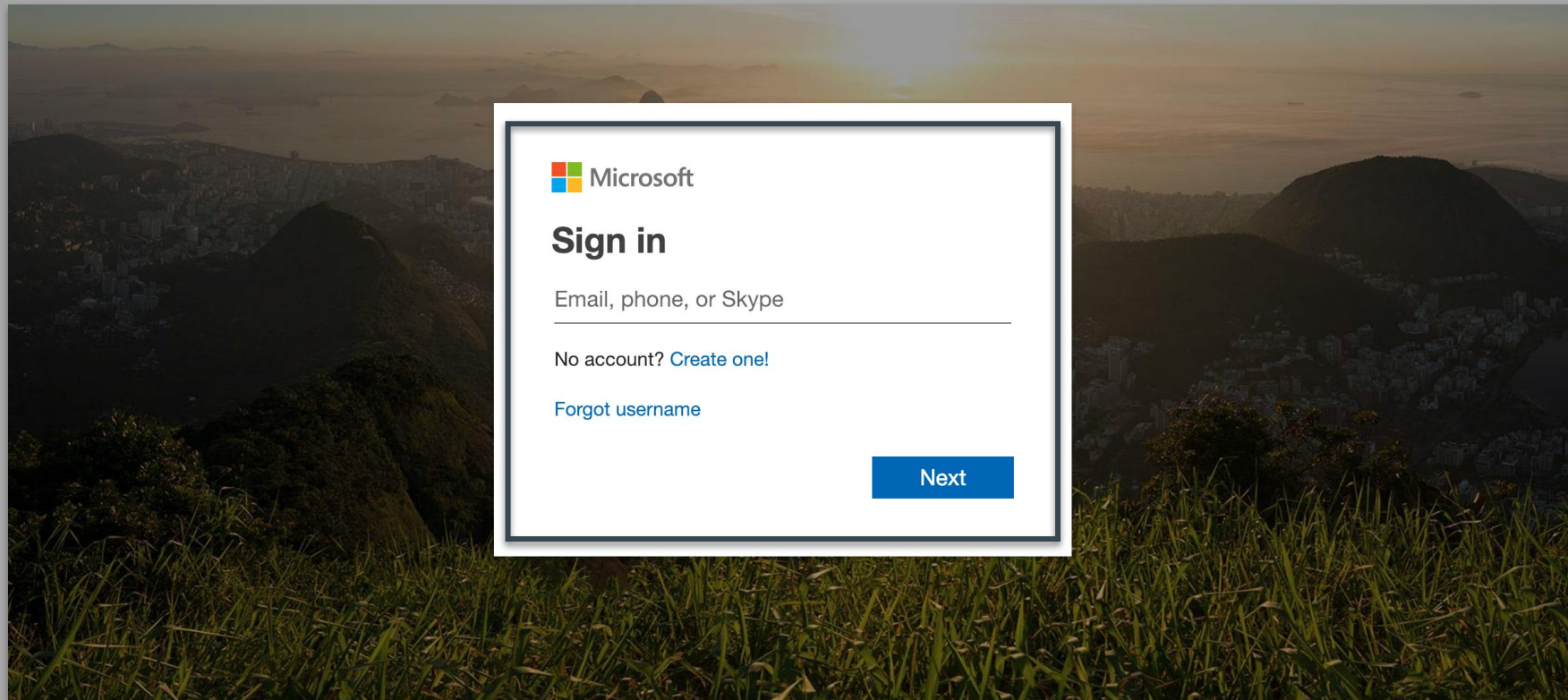
Sign in

First time Bing Maps developer?

To get started developing with Bing Maps, you will need a Bing Maps key. To create a key:

1. [Sign in](#) to the Bing Maps Dev Center with your existing Microsoft Account or create a new one.
2. Or, if you're an Azure customer, you can add Bing Maps to your Azure subscription through the [Azure Marketplace](#).

マイクロソフトのアカウントでログイン



この画面が出た人は一旦待機！！

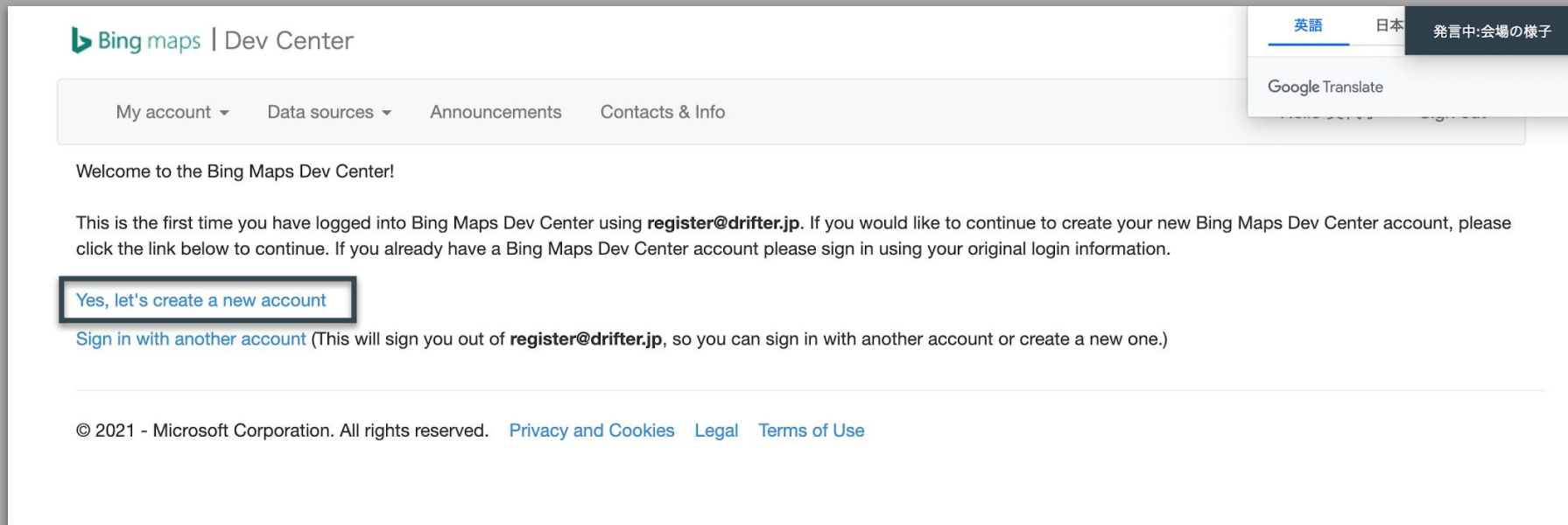
Announcement: No current service announcements.

To stay up to date on all the recent Bing Maps Platform releases and news, please visit the [Bing Maps Blog](#).

Important reminder regarding Bing Maps service notifications:

To ensure your company receives important Bing Maps service notifications and announcements that may affect your service availability, please make sure you always keep your organization's email contacts up-to-date in the 'Account Details' section of the Bing Maps Dev Center. **We recommend having multiple email address contacts listed there and the use of a distribution group email address is a suggested best practice (i.e.: bingmaps@contoso.com).**

こちらの画面が出た人は「Yes」-> アカウント作成！



Bing maps | Dev Center

英語 日本 発言中:会場の様子

My account ▾ Data sources ▾ Announcements Contacts & Info

Google Translate

Welcome to the Bing Maps Dev Center!

This is the first time you have logged into Bing Maps Dev Center using **register@drifter.jp**. If you would like to continue to create your new Bing Maps Dev Center account, please click the link below to continue. If you already have a Bing Maps Dev Center account please sign in using your original login information.

Yes, let's create a new account

[Sign in with another account](#) (This will sign you out of **register@drifter.jp**, so you can sign in with another account or create a new one.)

© 2021 - Microsoft Corporation. All rights reserved. [Privacy and Cookies](#) [Legal](#) [Terms of Use](#)

ここから全員共通

「My Keys」をクリック

Bing maps | Dev Center

My account ▾

Data sources ▾

Announcements

Contacts & Info

(macchinetta.xyz@gmail.com)

[Sign out](#)

Account Details

My Keys

Reports

Announcement: No current service announcements.


To stay up-to-date on the recent Bing Maps Platform releases and news, please visit the [Bing Maps Blog](#).

Important reminder regarding Bing Maps service notifications:

To ensure your company receives important Bing Maps service notifications and announcements that may affect your service availability, please make sure you always keep your organization's email contacts up-to-date in the 'Account Details' section of the Bing Maps Dev Center. **We recommend having multiple email address contacts listed there and the use of a distribution group email address is a suggested best practice (i.e.: bingmaps@contoso.com).**

© 2019 - Microsoft Corporation. All rights reserved. [Privacy and Cookies](#) [Legal](#) [Terms of Use](#)

アプリケーション名を入力

 | Dev Center

My account ▾

Data sources ▾

Announcements

Contacts & Info

(macchinetta.xyz@gmail.com)

[Sign out](#)

My keys

Create key

Application name *

Application URL


Key type * [What's This](#)

Basic ▾

Application type *

Dev/Test ▾

「Show key」をクリックするとAPIキーが表示される

 | Dev Center


My account ▾Data sources ▾AnnouncementsContacts & Info

(macchinetta.xyz@gmail.com)Sign out

My keys

Click [here](#) to create a new key.

Click [here](#) to download complete list of keys.

View Specific Key: 

Application name	Key details
20180919map_test	<div><div>Key: Show key</div><div>Application Url:</div><div>Key type: Basic / Dev/Test</div><div>Created date: 09/19/2018</div><div>Expiration date: None</div><div>Key Status: Enabled</div><div>Security Enabled: No</div></div> <div>Update Copy key Usage Report Enable Security</div>

現在地の取得

現在地の位置情報取得

種類

- `getCurrentPosition` : 実行時に現在地を取得する.
- `watchPosition` : 実行したら位置情報を取得し続ける.
- (`watchPosition`)はバッテリー消費に注意！！

オプション

- `enableHighAccuracy` : 対応端末でGPSを使用
- `maximumAge` : 指定時間以内であれば前回の値を用いる
- `timeout` : タイムアウト時間を設定

時間はミリ秒で指定する！

現在地の位置情報取得

```
<script>  
    // ()内には「成功時の関数」「失敗時の関数」「オプション」の順に記述する.  
  
    navigator.geolocation  
        .getCurrentPosition(showPosition, showError, option);  
</script>
```

現在地の位置情報取得

// 成功時の関数（位置情報を表示する）

```
function showPosition(position) {  
  const lat = position.coords.latitude;  
  const lng = position.coords.longitude;  
  console.log(lat, lng);  
}
```

まずはconsoleに表示！

現在地の位置情報取得

// 失敗時の関数（エラーを表示する）

```
function showError(error) {  
    let e = '';  
    if (error.code == 1) {  
        e = '位置情報が許可されてません';  
    } else if (error.code == 2) {  
        e = '現在位置を特定できません';  
    } else if (error.code == 3) {  
        e = '位置情報を取得する前にタイムアウトになりました';  
    }  
    alert('error: ' + e);  
}
```

エラーは3種類！

現在地の位置情報取得

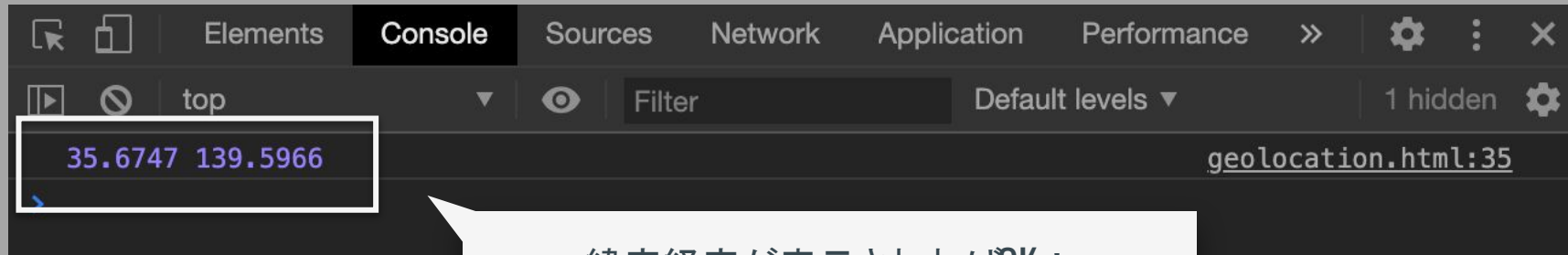
```
// オプション  
// オブジェクトの形
```

```
const option = {  
  enableHighAccuracy: true,  
  maximumAge: 20000,  
  timeout: 1000000,  
};
```

```
// この下に位置情報取得の処理 (navigator.geolocation...) を記述しよう！
```

現在地を取得して表示しよう！

実行結果



位置情報のデータの詳細(positionの中身)

The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. The top toolbar includes icons for navigation, a filter input, and a 'Default levels' dropdown. The console log shows a message from `geolocation.html:35` with an information icon. The log entry is a `GeolocationPosition` object. The `coords` property is expanded, showing `GeolocationCoordinates` with various properties. The `latitude` and `longitude` properties are highlighted with a white box. A white speech bubble with the text '緯度と経度!' (Latitude and Longitude!) points to this box. Below the highlighted values, the `speed` property is shown as `null`. Further down, the `__proto__` property is shown as `GeolocationCoordinates`, and the `timestamp` property is shown as `1592494540079`. At the bottom, the `__proto__` property is shown as `GeolocationPosition`.

```
geolocation.html:35
▼ GeolocationPosition {coords: GeolocationCoordinates, timestamp: 1592494540079} ⓘ
  ▼ coords: GeolocationCoordinates
    accuracy: 100
    altitude: null
    altitudeAccuracy: null
    heading: null
    latitude: 35.6747
    longitude: 139.5966
    speed: null
    ▶ __proto__: GeolocationCoordinates
  timestamp: 1592494540079
  ▶ __proto__: GeolocationPosition
```

緯度と経度！

位置情報取得の練習 (geolocation.html)

下記の処理を実装しよう！

- 読み込み時に位置情報を取得して`console.log()`に表示しよう！
- できる人は取得した緯度と経度をブラウザに表示しよう！
- （位置情報の取得までに時間がかかることもあるので気長に待つ！）

マップを表示する！

必要なファイルを読み込む

```
<script src='https://www.bing.com/api/maps/mapcontrol?mkt=ja-jp&key=API_KEY'></script>  
<script>                                     ▲自分のAPIキーを記述▲  
  // jsの処理  
</script>
```

処理の流れ

位置情報の取得

- navigator.geolocation...
- 成功時の関数, 失敗時の関数, オプションは先程と同様.
- (成功時の関数にマップ表示の処理を追加する)

マップの表示

- ブラウザに地図を表示する.
- 中心座標の指定が必要.
- (取得した緯度と経度を使用する)

準備

```
// マップ表示に使用する変数を用意しておく.  
// 失敗時の関数とオプションをコピーペーストしておく  
// （エラー時の警告とオプション内容は位置情報取得の処理と同様のため）
```

```
let map;
```

```
const option = {...};
```

```
function showError(error) {...}
```

地図の表示

```
// 位置情報取得に成功したときの関数 (mapsInit) を定義する.  
// 「地図を表示したいHTML要素のid」 「center」 「zoom」を指定する.  
  
function mapsInit(position) {  
    const lat = position.coords.latitude;  
    const lng = position.coords.longitude;  
    map = new Microsoft.Maps.Map('#map', {  
        center: {  
            latitude: lat, longitude: lng  
        },  
        zoom: 15,  
    });  
}
```


位置情報の取得

```
// 外部ファイルの読み込みがあるため、読み込み終了後に実行するよう指定する。  
// 他のファイルの読み込みが終わったら{}内を実行する  
// 現在地を取得し、成功したら地図を表示する関数を実行する  
window.onload = function () {  
    navigator.geolocation  
        .getCurrentPosition(mapInit, showError, option);  
}
```

【Point】

読み込み → 位置情報取得 → マップ表示 の順番

map表示の練習 (geo_and_map.html)

下記の処理を実装しよう！

- 位置情報の取得に成功したときに関数を定義し，map表示の処理を記述
- 位置情報を取得する処理

現在地が中心のmapが表示されればOK！

オリジナルマップ



【おまけ】ピンを立ててみる！

処理の流れ

ピンの生成

- ピンを立てる関数を定義する.
- ピンの色などを指定する.

マップ上に設置

- マップ表示後にピンを立てる関数を実行する.

ピンの生成

```
// 緯度, 経度, 表示したいマップを入力するとピンを立てる関数を定義.  
// ピンの色と表示するよう設定.
```

```
function pushPin(lat, lng, map) {  
  const location = new Microsoft.Maps.Location(lat, lng)  
  const pin = new Microsoft.Maps.Pushpin(location, {  
    color: 'navy',      // 色の設定  
    visible: true,      // これ書かないとピンが見えない  
  });  
  map.entities.push(pin);  
};
```

設定したピンをmap上に配置

// マップを表示する「mapsInit()」関数内でピンを設置する関数を実行.

```
function mapsInit(position) {  
  const lat = position.coords.latitude;  
  const lng = position.coords.longitude;  
  // マップ表示の処理 (省略)  
  pushPin(lat, lng, map);  
}
```

現在地にピンが立てばOK！

オリジナルマップ



【おまけ2】infoboxを出してみる！

処理の流れ

infoboxの生成

- infoboxを生成する関数を定義する.
- 表示内容を指定する.

マップ上に設置

- マップ表示後にinfoboxを生成する関数を実行する.

infoboxの生成

```
// 緯度, 経度, 表示したいマップを入力するとinfoboxをつくる関数を定義.  
// 表示内容を設定する.
```

```
function generateInfobox(lat, lng, map) {  
  const location = new Microsoft.Maps.Location(lat, lng)  
  const infobox = new Microsoft.Maps.Infobox(location, {  
    title: 'イマココ',  
    description: "I'm here!!!"  
  });  
  infobox.setMap(map);  
}
```

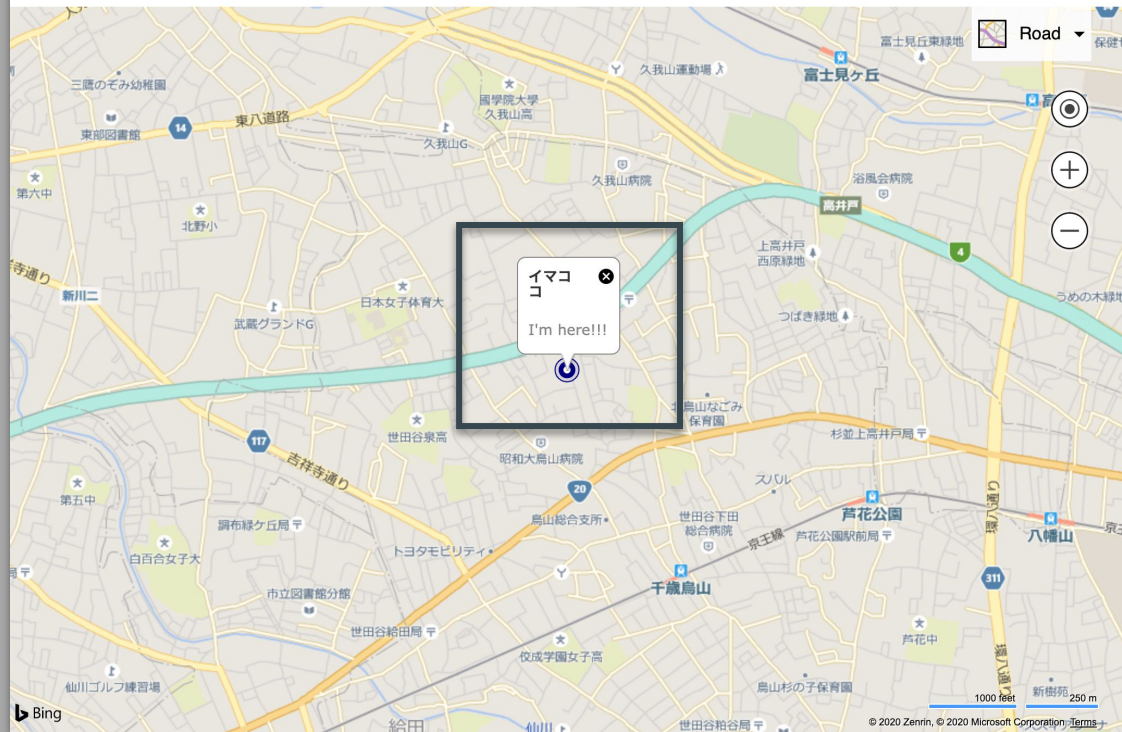
infoboxをmap上に配置

```
// マップを表示する「mapsInit()」関数内でinfoboxを生成する関数を実行.
```

```
function mapsInit(position) {  
  const lat = position.coords.latitude;  
  const lng = position.coords.longitude;  
  // マップを表示する処理 (省略)  
  pushPin(lat, lng, map);           // ピンを立てる関数  
  generateInfobox(lat, lng, map); // infoboxを生成する関数  
}
```

現在地にinfoboxが出ればOK！

オリジナルマップ



もう少しわしく！！

他にも機能たくさん！！

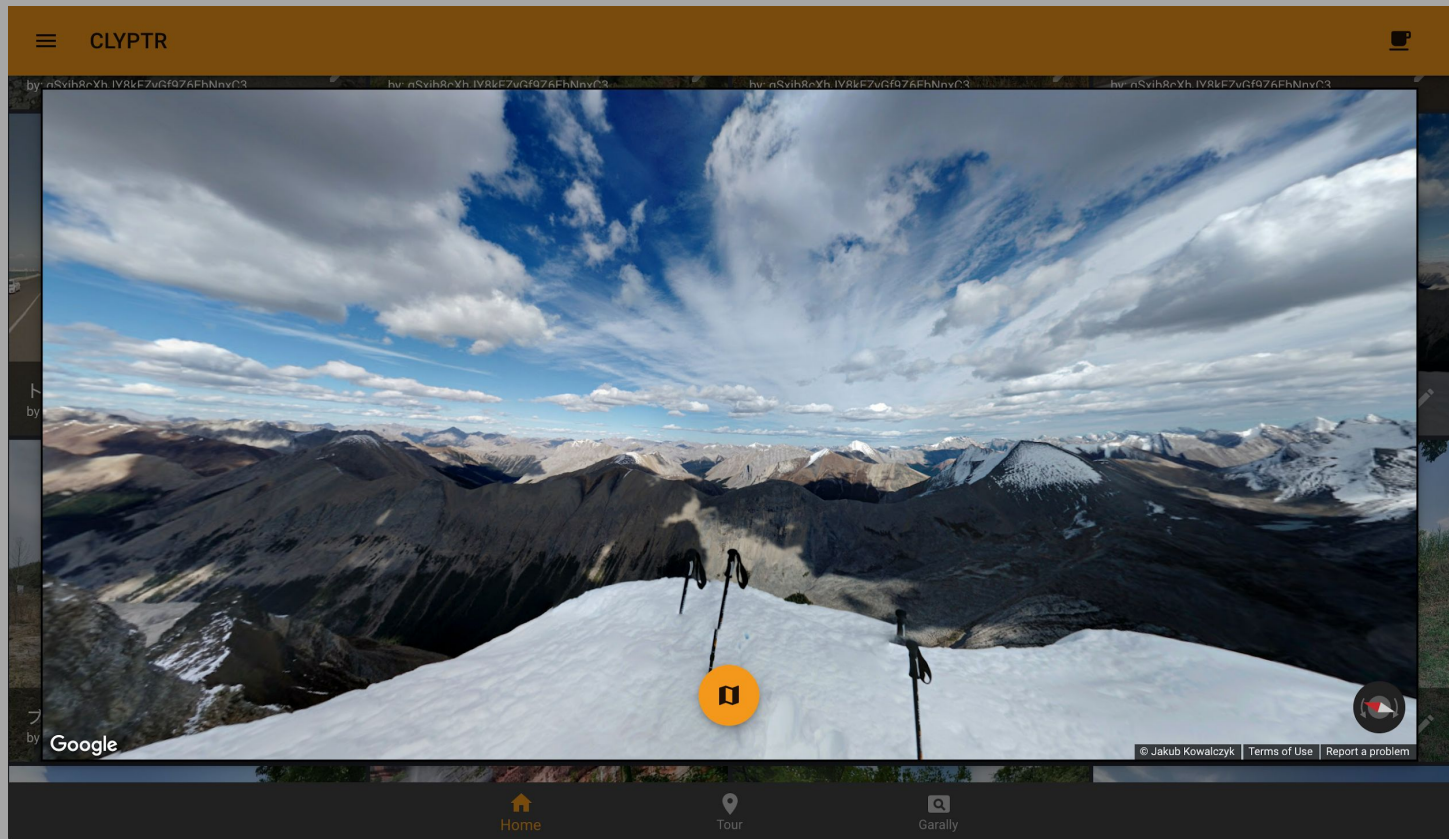
- ドキュメントも読んでみよう！
- <https://bingmapsv8samples.azurewebsites.net>

山崎先生が作ったサンプル集！！

- （かなり参考になる！！）
- <https://mapapi.org>

他のAPIの紹介

ストリートビュー (Google)



画像認識API (Google)



```
{  
  "mid": "/m/07030",  
  "description": "Sushi",  
  "score": 0.84657246,  
  "topicality": 0.846572  
},
```

翻訳API(いろいろ)

プログラミング楽しい！



```
{  
  "translatedText":  
    "!متعة البرمجة",  
  "detectedSourceLanguage":  
    "ja"  
}
```

レストラン検索API(ぐるなび)

検索条件を指定
(例：緯度経度と電子マネー可)

```
latitude=43.0656865  
&longitude=141.3693016  
&e_money=1
```



```
"rest": [  
  {  
    "@attributes": {  
      "order": 0  
    },  
    "id": "h308020",  
    "name": "ナガヤマ レスト",  
    "name_kana": "ナガヤマレスト",  
    "latitude": "43.065905",  
    "longitude": "141.364493",  
    "category": "和洋折衷喫茶",  
  },  
]
```

API紹介サイト

API First, ランキング

ログイン | 会員登録 | ? Help

API名を入力してください

APIbank
an AOS company

APIbankについて

API一覧

APIを使う・作る

APIblog

SOBAプラットフォーム

APIを一元管理し、動作環境を提供するAPI開発プラットフォーム

The logo for SOBA Platform features a stylized 'S' and 'O' in blue. To the right of the logo are several white line-art icons on a dark blue background: a magnifying glass, a document with a checkmark, a smartphone, a pencil, a ruler, a keyboard, and a gear.

課題

APIを使ったオリジナルのアプリケーションを実装

- 「API」を使用したアプリケーションを実装！！
 - APIを組み合わせる！
 - もちろん他の技術を組み込んでもOK！
 - API自体を作ってもいいぞ...！
- アレンジの例
 - ゾンビと対戦するマップ
 - ワンタップで周辺のレストランを検索
 - セカイのどこかへストリートビュー
 - 政府の統計情報からマーケット分析
 - twitter bot, LINE bot, etc...

まず自分の企画で使えそうなAPIを探せ！！

締切は厳守！！

P2Pタイム

まずはチーム内で解決を目指す！

訊かれた人は苦し紛れでも応える！！