

交通指引

使用Google Maps Javascript API

1. 首先到Google Cloud Platform創立專案
2. 根據需要使用的服務，查詢相對應的API，申請後就會顯示在自己的Platform畫面，會顯示request次數
3. 將API key放入html的body中



獲取路線規劃、顯示最佳路徑&時間、切換交通方式：**Directions API**

縣市鄉鎮資料：**Geocoding API**

取得路線時間、距離：**Distance Matrix API**

Directions API

refer:

1. <https://github.com/sherwyn11/KEDS-Rideshare-KUHacks/blob/master/client/src/views/Maps/Maps.js>
2. <https://icelandcheng.medium.com/使用google-map-api-directions-service-獲取及顯示最佳路徑-cf82060b90f2>

```
import {
  withScriptjs,
  withGoogleMap,
  GoogleMap,
  DirectionsRenderer,
} from "react-google-maps";

const Maps = compose(
  withProps({
    googleMapURL:
      "https://maps.googleapis.com/maps/api/js?key=Your API Key&libraries=geometry,drawing,places",
    loadingElement: <div style={{ height: `100%` }} />,
    containerElement: <div style={{ height: `580px` }} />,
    mapElement: <div style={{ height: `100%` }} />
  }),
  withScriptjs,
  withGoogleMap,
```

place在libraries那邊如果使用API的話一定要用，drawing跟geometry則分別代表繪圖跟定位

```
import { compose, withProps, lifecycle } from "recompose";
lifecycle({
  componentDidMount() {
    const DirectionsService = new google.maps.DirectionsService();

    DirectionsService.route(
      {
        origin: new google.maps.LatLng(19.1217707406339, 72.83944134193857),
        destination: new google.maps.LatLng(19.17060002212831, 72.79517092313263),
        travelMode: google.maps.TravelMode.DRIVING
      },

      (result, status) => {
        if (status === google.maps.DirectionsStatus.OK) {
          this.setState({
            directions: result,
            distance: result.routes[ 0 ].legs[ 0 ].distance.text,
            time: result.routes[ 0 ].legs[ 0 ].duration.text,
          });
        } else {
          console.error(`error fetching directions ${result}`);
        }
      }
    );
  },
});
```

1. 使用google.maps.DirectionsService()new一個物件DirectionsService
2. 用route()這個方法傳入對應的參數，可傳的參數如下：

```
{
  origin: LatLng | String | google.maps.Place,
  destination: LatLng | String | google.maps.Place,
  travelMode: TravelMode,
  transitOptions: TransitOptions,
  drivingOptions: DrivingOptions,
  unitSystem: UnitSystem,
  waypoints[]: DirectionsWaypoint,
  optimizeWaypoints: Boolean,
  provideRouteAlternatives: Boolean,
  avoidFerries: Boolean,
  avoidHighways: Boolean,
  avoidTolls: Boolean,
  region: String
}
```

origin(必傳)：起點資訊，傳的格式可以用經緯度、地址或者google.maps.Place的資訊，例如也可以傳 龍山寺。

destination(必傳)：終點資訊，傳的格式可以用經緯度、地址或者google.maps.Place的資訊。

travelMode(必傳)：從起點到終點移動的方式，包括：**BICYCLING(腳踏車)**、**DRIVING(開車)**、**TRANSIT(轉運)**、**WALKING(走路)**等四種，如果設定是**DRIVING**的話，可以另外傳送**drivingOptions**，設定交通時間計算模式，後面會再介紹**drivingOptions**，另外比較特別的是**TRANSIT**，如果傳送這個設定，表示起點跟終點間是有轉運點的，像搭飛機會在某個地方轉機那樣，不會直接抵達終點，所以如果設定**TRANSIT**的話，還會需要另外傳送**transitOptions**參數，下面會介紹**transitOptions**。

transitOptions(選填)：**travelMode**是**TRANSIT**時才需要傳送，可填的選項包括**arrivalTime(抵達時間)**、**departureTime(出發時間)**、**modes(中繼點，有像是RAIL、SUBWAY、TRAIN等選項)**、**routingPreference(路徑偏好，可以設定FEWER_TRANSFERS、LESS_WALKING這種轉運方式的偏好)**。

drivingOptions(選填)：travelMode是DRIVING時才需要傳送，可傳送的參數有departureTime(出發時間)及trafficModel(交通時間計算模式，有bestguess(最佳模式)、pessimistic(悲觀模式)、optimistic(樂觀模式)等三個模式來計算到達終點所需時間，預設值為bestguess，bestguess是用最貼近實際交通狀況來估計時間，pessimistic是用最糟的情況來預估時間而optimistic適用最好的情況來預估，所以一般情況，得到的時間長度應該是：pessimistic > bestguess > optimistic)

unitSystem(選填)：距離單位，有google.maps.UnitSystem.METRIC及google.maps.UnitSystem.IMPERIAL兩個選項，沒有傳送unitSystem參數的話，預設為METRIC，也就是公制，得到的距離單位會是公尺、公里，如果設定是IMPERIAL，回傳值的距離單位就會是英尺、英里。

waypoints(選填)：路徑中間是否有經過哪些特定的地點，可以傳送多個地點，每個地點要傳送的參數包括location(點位資訊，傳的格式可以用經緯度、地址或者google.maps.Place的資訊)及stopover(是否停留，傳送格式為Boolean，若為TRUE，則回傳的路徑會被此點為切分為兩條路徑)。

optimizeWaypoints(選填)：若有傳送waypoints資訊，則可傳送TRUE or FALSE 來設定是否讓Directions Service決定路徑中的waypoints的最佳順序。

provideRouteAlternatives(選填)：是否提供多個路徑的資訊，可傳送TRUE or FALSE，若傳送TRUE，則Directions Service會回傳多個建議的路徑資訊，不會只回傳一個最佳路徑資訊。

avoidFerries(選填)：是否避開渡輪，可傳送TRUE or FALSE，要求Directions Service回傳的結果是否要避開使用渡輪的路徑。

avoidHighways(選填)：是否避開高速公路，可傳送TRUE or FALSE，要求Directions Service回傳的結果是否要避開高速公路的路徑。

avoidTolls(選填)：是否避開收費站，可傳送TRUE or FALSE，要求Directions Service回傳的結果是否要避開收費站的路徑。

region(選填)：限定回傳結果是要特定區域顯示的資訊，可傳送像是GB, US這樣的地區代碼，因為Google Map在不同地區顯示的結果可能會有些微不同，例如在 <https://maps.google.com/> (the United States)跟在 <http://maps.google.es/> (Spain)搜尋"San Francisco"結果可能會不一樣，要查詢各地區應該傳送的代碼可參考這邊。

3. 如果status顯示ok，則將結果儲存起來

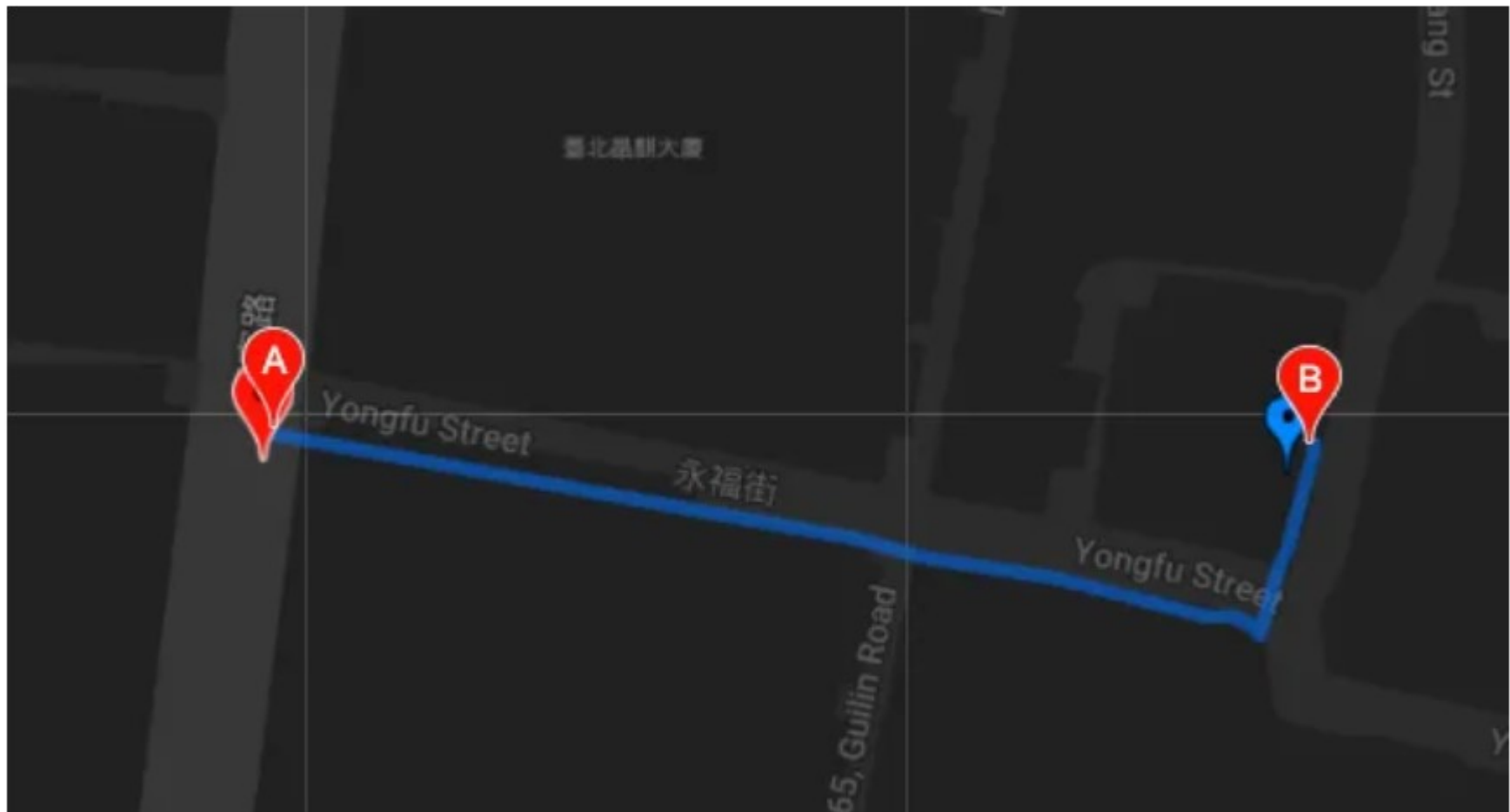
拿HTML來看看DirectionsRenderer()將路徑顯示在地圖的結果

以下圖為例，設定所在位置為起點，咖啡廳為終點，移動方式為走路



只要把得到的response結果帶入directions這個參數，然後帶入要畫的地圖object給map參數，在整包hash傳給DirectionsRenderer()就可以把路徑畫出來了

```
let directionsService = new google.maps.DirectionsService();
let request = {
  origin: haight,
  destination: oceanBeach,
  travelMode: 'WALKING'
};
directionsService.route(request, function(response) {
  if (status == 'OK') {
    let directionsDisplay = new google.maps.DirectionsRenderer({
      map: map,
      directions: response,
    });
  }
});
```



路徑預設樣式