Balazs Barcza

SB18002

Mobile Development - HDIP Feb 2018 - PT

Project Title: Travel Pal

Table of Contents

Mobile Development - HDIP Feb 2018 - PT....1
Section 1: Assignment Introduction....3
Section 2: Technologies....3
Section 3: Specific Requirements....3
Section 4: Design Process....4
Section 5: Code and App....5
Section 5: Adobe cloud compiling process....16

Section 1: Assignment Introduction

A company has come to you as a developer and asked you to develop a mobile application that will work on multiple platforms. They know it is a very difficult task to install all the different libraries and SDKs to do this, so they have told you that using a Cloud based compiling tool is the best option. Design and build a mobile application that will use the geolocation of your phone to show the city and country where you are, the weather at the current location, and the currency exchange rate to the local money.

Section 2: Technologies

Adobe Phonegap
– PhoneGap is a tool which allows you to write applications in HTML and JavaScript and deploy them to any phone.
Front end JavaScript Framework
– You can use any HTML/CSS/JavaScript based user interface library such as jQuery Mobile, Framework7 or Bootstrap.
External APIs
– In order to be able to access real time data, you will need to connect to external API to request weather, currency exchange rate and name of the city/country where you are located. You could use the APIs used in class, or explorer different options.

Section 3: Specific Requirements

Start with a template source code, the following modifications should be made:

• The application should have a welcome screen that displays the name of the city/country based on the data collected by the GPS sensor of the phone.

• A section should exist in the application that allows the user to enter in a certain amount of USD to convert to the local currency and viseversa.

• Another section should exist in the application to check the weather of the current location of the user.

• Finally, there must be button for the user to save automatically the current location and data collected from the different API’s to be retrieved later, this way the user can keep record of places where they have been.

• This is an individual assignment.

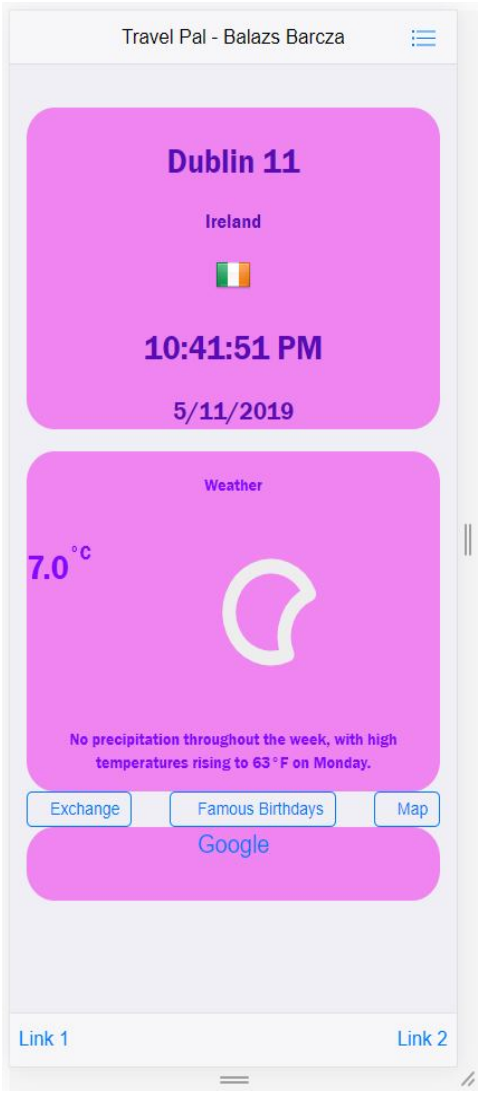
Section 4: Design Process

Plan your day like magic

The Travel-Pal makes it easier than ever to plan and organize your trips. It automatically maps your location, It is suggestions for location. If you want to exchange your money you can find a page where you can change your USD dollar to the local currency and back way. The home page gets the local weather and local time. I have a page where you can find all of the famous people who born this day and extra they are age.

Section 5: Code and App

This is the welcome screen. It is a clear design. The most dominant color is the ping I wanted to make the design user-friendly and I am using block layout.



On the **welcome screen**, You can see the location base time. It is updating every 1 second.

```
function getTime( ){
var d = new Date();
var d1 = new Date().toLocaleTimeString(); // 11:18:48 AM
document.getElementById("gettimes").innerHTML = d1;
//document.getElementById("getminutes").innerHTML = d.getMinutes();
//document.getElementById("gettime").innerHTML = d.getHours();

var today = new Date().toLocaleDateString();
// 07-06-2016 06:38:34
//var dd = String(today.getDate()).padStart(2, '0');
//var mm = String(today.getMonth() + 1).padStart(2, '0'); //January is 0!
//var yyyy = today.getFullYear();

//today = mm + '/' + dd + '/' + yyyy;
document.getElementById("getmount").innerHTML = today;

setTimeout(getTime, 1000);
}
```

I have the main javascript file: my-app.js. I have used only js file I had an issue with split the code.

You can see the code belong It is showing function, It is getting location from the device.

```
// global variable
var Latitude;
var Longitude;
var contryToExchang;
var currencyCODE;

function sendLocation(){
    var lat =Latitude;
    var ltu = Longitude;
    getLocationMain(lat,ltu);
}

// geoLocation function
function geoLocation(){

    navigator.geolocation.getCurrentPosition(geoCallback, onError)


}


// onSuccess Callback
    //   This method accepts a `Position` object, which contains
    //   the current GPS coordinates
    //
    function geoCallback(position) {
        console.log(position);
        /*
        var element = document.getElementById('geolocation');
        element.innerHTML = 'Latitude: '  + position.coords.latitude      + '<br />' +
                            'Longitude: ' + position.coords.longitude     + '<br />' +
                            '<hr />'      + element.innerHTML;
        */
                            // pass the location to opencage

                           Latitude =position.coords.latitude;
                           Longitude =position.coords.longitude;
                            console.log('Latitude: '+Latitude);
                            console.log('Longitude: '+Longitude);
                            openCage();
                            weatherMain(Latitude,Longitude);
                            GEOweather(Latitude,Longitude);
```

The next code getting the flag.

```
contryToExchange = responseJSON.results[0].components.country;
var currencyCODE = responseJSON.results[0].annotations.currency.iso_code;
console.log("Contry code here!!!" + countryCode);
document.getElementById('littleFlag').src = "https://www.countryflags.io/" +
countryCode + "/shiny/32.png";
```

https://www.countryflags.io/

This API is really simple you just have to copy the code to the HTML file.

```
<img
src="https://www.countryflags.io/:country_code/:style/:size.png">
```

`country_code:` The app just passed the country location code: IRL / HUF / USD /GBP

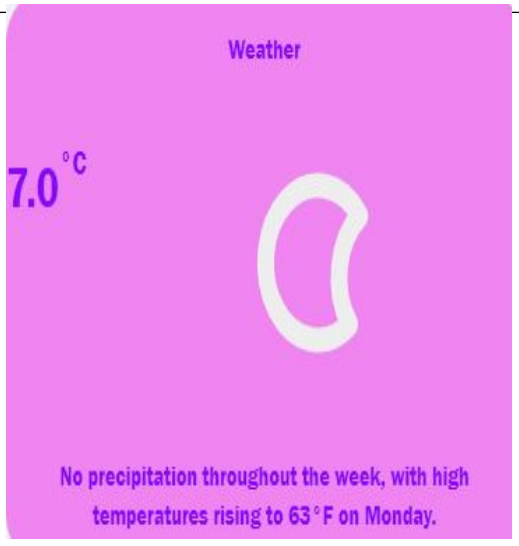
Weather API:
https://darksky.net
The Dark Sky Company specializes in weather forecasting and visualization. Back in 2011, we had the crazy idea that robots could predict the weather with down-to-the-minute precision, and thanks to a handful of generous strangers, we were able to give it a shot. Since then, those robots have become "scarily accurate," powering our own award-winning weather app, Dark Sky, in addition to thousands of other businesses, apps, and crazy ideas.

For a software startup, we're a bit unusual: we're self-funded, have been profitable from the start, and funnel a lot of our efforts into researching new ways to do what we do. Although we're no longer a two-man operation—having partnered with a company called Applied Invention and grown our team—our focus has always been on sustainability over growth, and we plan to keep it that way.

If you have any questions about us, our app, or about the weather in general, please feel free to drop us a line: we'd love to hear from you.

Adam, Jay, and the Dark Sky Team
Cambridge, MA

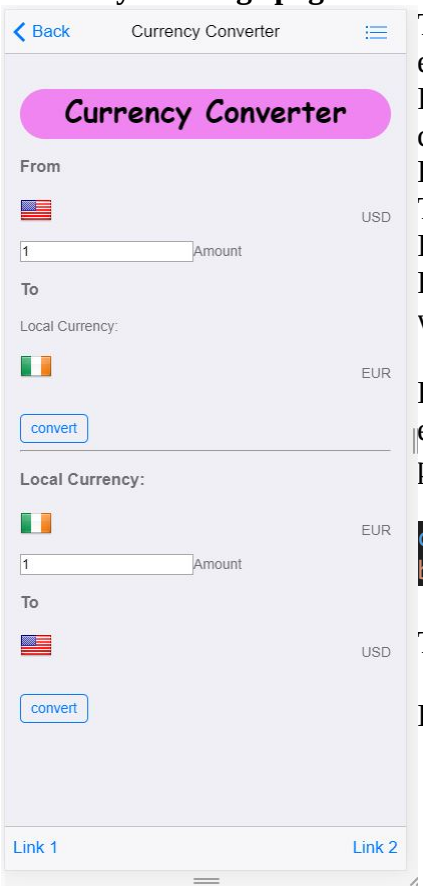
```
function weatherMain(Latitude,Longitude) {
    var Latitude = Latitude;
    var Longitude = Longitude;
    var http = new XMLHttpRequest();
    //const url = 'https://api.darksky.net/forecast/3ad7f8e54c6fdcafbe0dfa539a9ae18c/37.8267,-122.4233';
    const url = 'https://api.darksky.net/forecast/3ad7f8e54c6fdcafbe0dfa539a9ae18c/' + Latitude + ',' + Longitude +'';

    http.open("GET", url);
    http.send();

    http.onreadystatechange = (e) => {

        // First, I'm extracting the reponse from the
        // http object in text format
        var response = http.responseText;

        // As we know that answer is a JSON object,
        // we can parse it and handle it as such
        var responseJSON = JSON.parse(response);

        // Printing the result JSON to the console
        console.log(responseJSON);
        var locationTimezone = responseJSON.timezone;
        console.log(locationTimezone);
        var temp = responseJSON.currently.temperature;
        console.log(temp);
        var celsius = Math.floor((temp - 32) * 5 / 9);

        var daily = responseJSON.daily.summary;  //Skycons
```




```
    var iconRequest = responseJSON.currently.icon;

    var icons = new Skycons({'color' : '#eeeeee'});

    var iconList = [
        "clear-day",
        "clear-night",
        "partly-cloudy-day",
        "partly-cloudy-night",
        "cloudy",
        "rain",
        "sleet",
        "snow",
        "wind",
        "fog"
    ];
    console.log(icons);
    for (i = 0; i < iconList.length; i++) {
        if (iconRequest == iconList[i]) {
                icons.set('icon', iconList[i]);

        }
    }
    icons.play();

//document.getElementById('location-timezone').innerHTML = locationTimezone;
document.getElementById('temperatureMain').innerHTML = celsius.toFixed(1);
document.getElementById('daily').innerHTML = daily;
```

Temperature and extra data about the weather, skycon is always change dependence the weather forecast	Skycons icon program lines

Currency exchange page:



This page gives data back from the server to check the local currency exchange rate.
If the user change country the flag is changed to the local flag and currency is automatically change as well. Hungary: HUF, Ireland: EUR.
The layout is simple easy to understand.
I am using in my program the https://exchangeratesapi.io/ API.
Foreign exchange rates API
with currency conversion

Exchange rates API is a free service for current and historical foreign exchange rates
published by the European Central Bank

```
const url = 'https://api.exchangeratesapi.io/latest?base=USD';
```

This information base on USD -dollar.

If I need other information I just have to change the base currency.

JSON:

{"base":"USD","rates":{"BGN":1.7415850401,"NZD":1.5160284951,"ILS":3.5622439893,"RUB":65.3154942119,"CAD":1.3474621549,"USD":1.0,"PHP":52.2902938557,"CHF":1.0131789849,"AUD":1.4300089047,"JPY":109.750667854,"TRY":6.1297417631,"HKD":7.8481745325,"MYR":4.1657168299,"HRK":6.5975066785,"CZK":22.9136242208,"IDR":14405.5031166518,"DKK":6.6480854853,"NOK":8.7438112199,"HUF":288.0854853072,"GBP":0.768032057,"MXN":19.1975957257,"THB":31.5796972395,"ISK":121.9946571683,"ZAR":14.2263579697,"BRL":3.9541406946,"SGD":1.3628673197,"PLN":3.8254674978,"INR":69.9977738201,"KRW":1180.160284951,"RON":4.2384683882,"CNY":6.8235084595,"SEK":9.6267141585,"EUR":0.8904719501},"date":"2019-05-10"}

program code:

```
function convert(){
geoLocationCC();
var http = new XMLHttpRequest();
const url = 'https://api.exchangeratesapi.io/latest?base=USD';
/*
different api
const url = 'http://apilayer.net/api/live?
access_key=310ff77de7a824ad7b6774e18cf4e29e&currencies=EUR,GBP,CAD,PLN&source=USD';
```

```
const url = 'http://apilayer.net/api/live?access_key=310ff77de7a824ad7b6774e18cf4e29e&currencies='+
currencyCODE+'&source=USD';
const url = 'https://api.exchangeratesapi.io/latest?symbols=HUF';
const url = 'https://free.currconv.com/api/v7/convert?q=EUR_HUF&compact=ultra&apiKey=8e9a85acc5ca01bfbb5e';
*/

console.log('What happining here'+currencyCODE);
http.open("GET", url);
http.send();
http.onreadystatechange = (e) => {
// First, I'm extracting the reponse from the
// http object in text format
var response = http.responseText;
// As we know that answer is a JSON object,
// we can parse it and handle it as such
var responseJSON = JSON.parse(response);
// Printing the result JSON to the console
console.log(responseJSON);
help = "EUR_"+currencyCODE;
console.log(help);
var rate = responseJSON.rates[currencyCODE];
console.log("exchange java :"+rate);
var input =document.getElementById('input').value;
console.log("INPUT here"+input);
var result = input * rate;
document.getElementById('result').innerHTML = result;
}
}
```

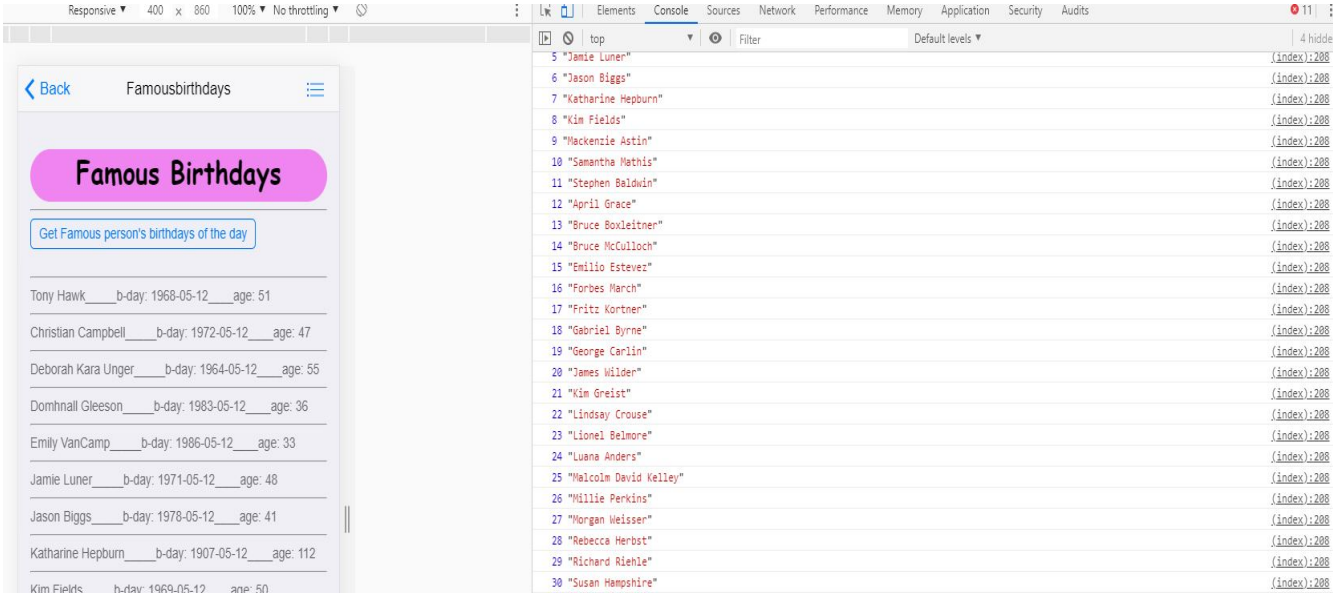
I could use different API but The European Central Bank was easy to integral to the program.
I did not need API key I have unlimited call.

```
/*
different api
const url = 'http://apilayer.net/api/live?
access_key=310ff77de7a824ad7b6774e18cf4e29e&currencies=EUR,GBP,CAD,PLN&source=USD';
const url = 'http://apilayer.net/api/live?access_key=310ff77de7a824ad7b6774e18cf4e29e&currencies='+
currencyCODE+'&source=USD';
const url = 'https://free.currconv.com/api/v7/convert?q=EUR_HUF&compact=ultra&apiKey=8e9a85acc5ca01bfbb5e';
*/
```

This is the console screenshot.

EUR_EUR(index):208
exchange java :0.8904719501

Famous birthday page:

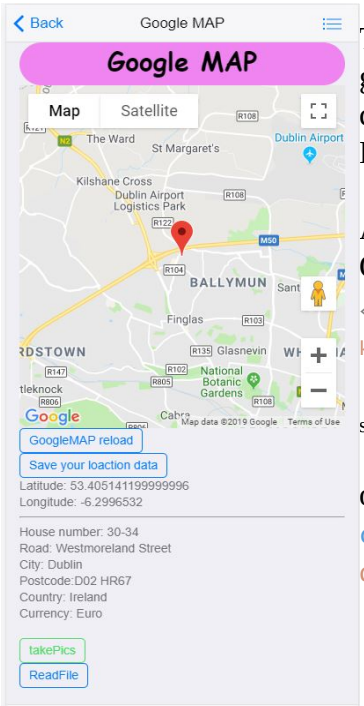


This page gives information about the famous person who born this day.
information name / born date / age
I have a little bug I always pull down 2 times the information If I have more time I would like to fix this issue. I think could be nice update If I can import Wikipedia information about the person or create a google search button. It can be the next version.

```
const url = 'https://celebritybucks.com/developers/birthdays/JSON';
```

API: https://celebritybucks.com/developers

Google Map page:



This page helps to check your location on the google map. You can get extra information about the latitude and longitude, and the user can save information a text file and display it.
Extra you can take a photo and display on the page.

API:
Google API:

```
<script src="https://maps.googleapis.com/maps/api/js?
key=AIzaSyApFPNQ2WxbEueUXMJBPHjLenlPnFo68ls&callback=initMap"></script>
```

script on the html file

opencagedata API:

```
const url = 'https://api.opencagedata.com/geocode/v1/json?
q=53.346+-6.2588&key=b95ccb4da5784bfca72b01a6b2af690f';
```

The code get information about the location.

```
function writeOutInfo(){

    var printOutLocation = document.getElementById('gpslocation');
    printOutLocation.innerHTML = 'Latitude: '  + Latitude      + '<br />' +
                                 'Longitude: ' + Longitude     + '<br />' +
                                 '<hr />'      + printOutLocation.innerHTML;

    var http = new XMLHttpRequest();
    const url = 'https://api.opencagedata.com/geocode/v1/json?q=53.346+-6.2588&key=b95ccb4da5784bfca72b01a6b2af690f';
    http.open("GET", url);
    http.send();
    http.onreadystatechange = (e) => {
        var response = http.responseText;
        var responseJSON = JSON.parse(response);

        console.log(responseJSON);

        house_number = responseJSON.results[0].components.house_number;
        road = responseJSON.results[0].components. road;
        city = responseJSON.results[0].components.city;
        postcode =responseJSON.results[0].components.postcode;
        country = responseJSON.results[0].components.country;
        var currency = responseJSON.results[0].annotations.currency.name;

        document.getElementById('getlocationC').innerHTML ="House number: "+house_number+"<br>Road: "+road+ "<br>City: " + city +"<br>Postcode:"+postcode+
                                                            "<br>Country: " + country +"<br>Currency: " + currency;
    }
```

Saving and reading the text file.

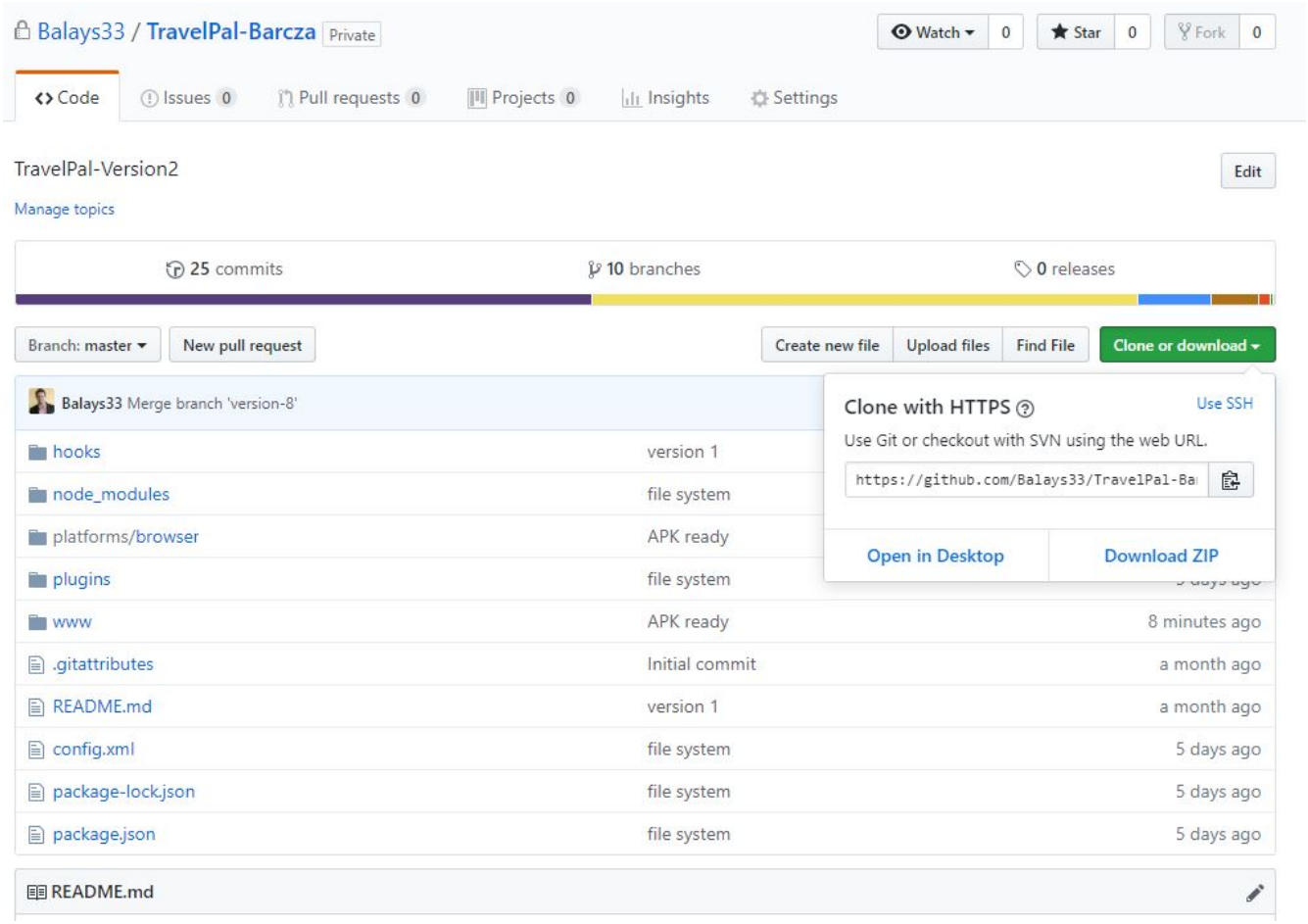
```
▼{documentation: "https://opencagedata.com/api", licenses: Array(1), rate: {…}, results: Array(1), status: {…}, …}
   documentation: "https://opencagedata.com/api"
  ▶licenses: [{…}]
  ▶rate: {limit: 2500, remaining: 2497, reset: 1557705600}
  ▼results: Array(1)
    ▼0:
      ▶annotations: {DMS: {…}, ITM: {…}, MGRS: "29UPV8246614269", Maidenhead: "IO63ui83wb", Mercator: {…}, …}
      ▶bounds: {northeast: {…}, southwest: {…}}
      ▶components: {ISO_3166-1_alpha-2: "IE", ISO_3166-1_alpha-3: "IRL", _type: "building", city: "Dublin", city_district: "Royal Exchange A ED", …
       confidence: 10
       formatted: "30-34 Westmoreland Street, Royal Exchange A ED, Dublin, County Dublin, D02 HR67, Ireland"
      ▶geometry: {lat: 53.3460446, lng: -6.2588287}
      ▶__proto__: Object
       length: 1
    ▶__proto__: Array(0)
  ▶status: {code: 200, message: "OK"}
  ▶stay_informed: {blog: "https://blog.opencagedata.com", twitter: "https://twitter.com/opencagedata"}
   thanks: "For using an OpenCage Data API"
  ▶timestamp: {created_http: "Sun, 12 May 2019 10:57:48 GMT", created_unix: 1557658668}
   total_results: 1
  ▶__proto__: Object
```

```
Saving My  Latitude: 53.405141199999996Longitude: -6.2996532                    (index):208
Successful file write...                                                        (index):208
Successful file read:  Latitude: 53.405141199999996                             (index):208
Longitude: -6.2996532
Currency code: EUR
 house_number: 30-34
road: Westmoreland Street
city: Dublin
postcode: D02 HR67
country: Ireland: Ireland
file path: /locationFile.txt                                                    (index):208
```

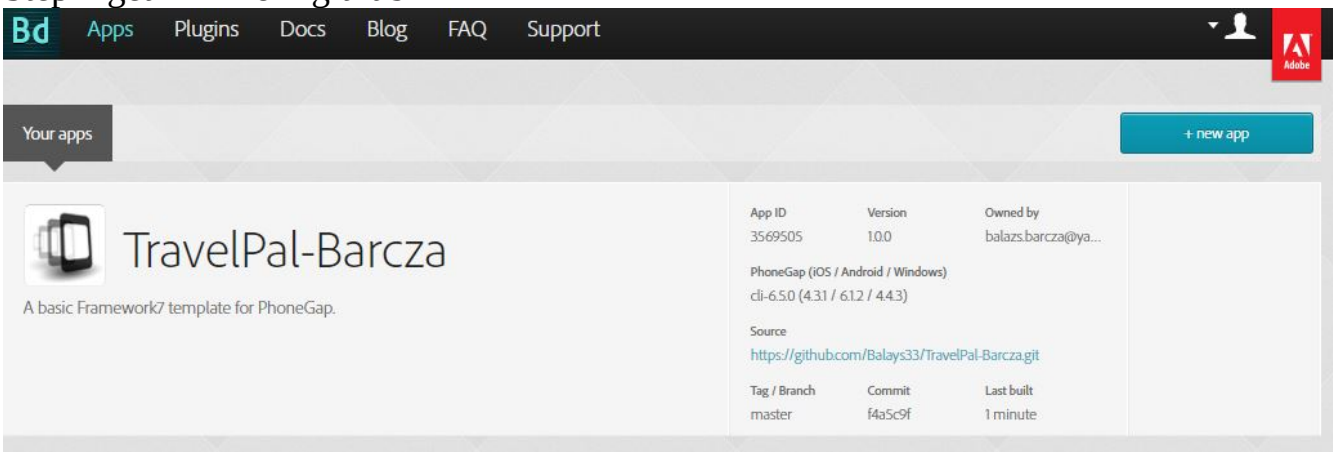
Section 5: Adobe cloud compiling process

o Document containing screenshots of your application in action, screenshots of the Adobe cloud compiling process, and justification for your design decisions (graphic and logic).

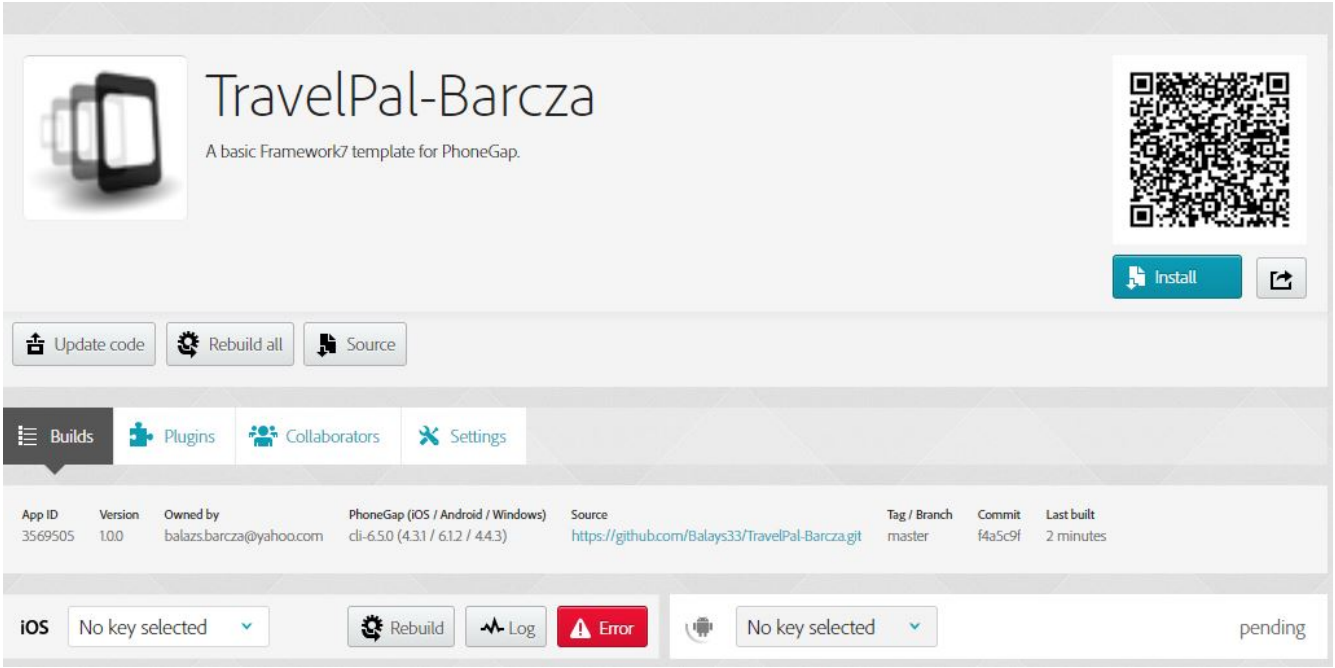
o Binary Android package which can be downloaded from the Adobe Cloud website once compiled (APK file).



Step 1 get link from github



Step 2 building phonegap project



Step 3 ready to test on mobile phone just hast to download the app