

Showing actions on a map using the REST API

Iain Ure - Frontend Developer

MAY 2021

What we will cover

- Demo
- The browser code
- How does it work
- The server
- The Engaging Networks profile
- The Engaging Networks export group
- ENS API call #1: Authentication
- ENS API call #2: Profile data
- Geocoding
- Implementation
- Q&A

Demo

Example

Browser code



```
document.addEventListener("DOMContentLoaded",function(){  
  
  // init the map  
  const mymap = L.map(mapId).setView([51.505, -0.09], 13)  
  
  // add our tile Layer  
  L.tileLayer(  
    `https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access_token={accessToken}`,  
    {  
      attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/copyright">O  
      maxZoom: 18,  
      id: 'mapbox/dark-v9',  
      tileSize: 512,  
      zoomOffset: -1,  
      accessToken: mapBoxToken  
    }  
  ).addTo(mymap)  
  
  loadAndApplyMarkers(mymap)  
  
})
```

(Mapbox tile layer free for the first 50,000 loads)

Browser code

```
// contacting our server and getting our Latitude and Longitude
async function loadAndApplyMarkers($map){

  try {

    // Load points from our server (which loads postcodes from EN and geocodes them)
    let {data} = await axios.get('http://localhost:3001/points')

    const $markers = []

    data.forEach(point => {

      const $marker = L.marker([point.latitude, point.longitude]).addTo($map)
      $marker.bindPopup(point.firstName)

      $markers.push($marker)

    })

    // zoom and position the map so all markers are in view
    const $markerGroup = new L.featureGroup($markers)
    $map.fitBounds($markerGroup.getBounds())

  }

  catch(err){

    alert(`Unable to load croco-data: ${err}`)

  }

}
```

Browser code

GET

`https://localhost:3001/points`

Response:

```
[
  {
    "latitude": "-26.2573212",
    "longitude": "28.0478826",
    "firstName": "Iain Ghana"
  },
  {
    "latitude": "48.8640493",
    "longitude": "2.3310526",
    "firstName": "Iain FR"
  }
]
```

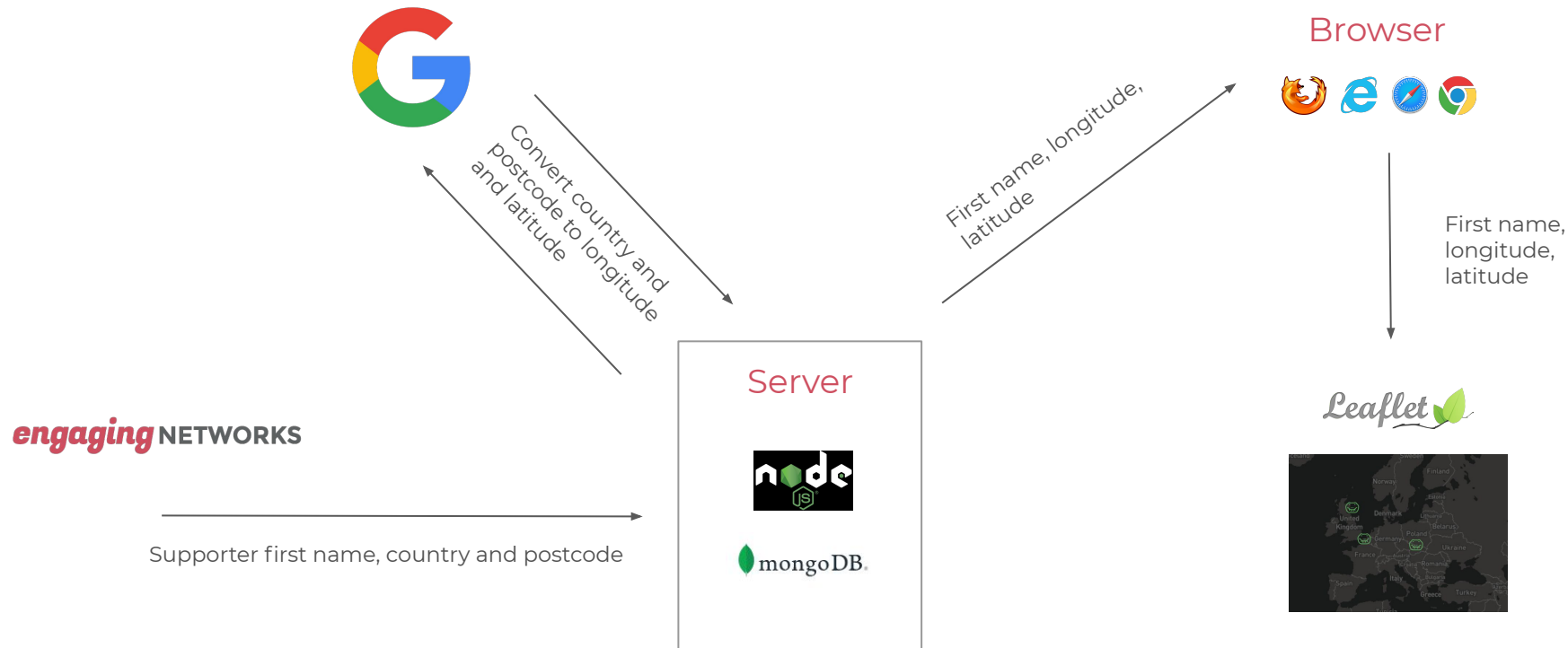
Google geocoding service

Engaging Networks

How does it work

Where is this data coming from?

How does it work



The Server

What does the server do?

1. Authenticate to ENS REST
2. GET supporter first name, country and postcode from ENS REST
3. Geocode that data

Why can't we just do this in the browser?

- Cannot make these ENS REST calls from the browser
 - No CORS
 - Allowed IPs only
 - Private token must be kept safe
- Google geocoding costs 0.005 USD each request
 - Keep our token secure otherwise anyone can use it
 - Store already-geocoded data
 - Need a database

The Server

Environment: Node JS

Database: Mongo DB

NPM packages

Webserver: [Express JS](#)

DB connection: [Mongoose](#)

API calls: [Axios](#)

Scheduling: [node-schedule](#)



The Server

Server code tour

The Profile

Data & reports / Profiles

- Segment of user data
- Has taken action
- Has opted in to being shown on a map

When does it get updated?

- Run manually
- Run automatically every night



Supporters will not appear immediately on the map.

[More on profiles](#)

Edit Profile

Name

Crocodiles action

Notes

Filter Conditions

☒ AND ☐ OR

Filters

Select a filter type

Add filter

Question Responses

Find supporters whose response to

Map optin ✕

Equals

Y

case insensitive

Use ~ between text for multiple values.

Data Capture Activity

Find supporters whose activity count is

greater than

0

In these pages

ENCC2021 Crocodiles (do not touch) ✕

Since this date

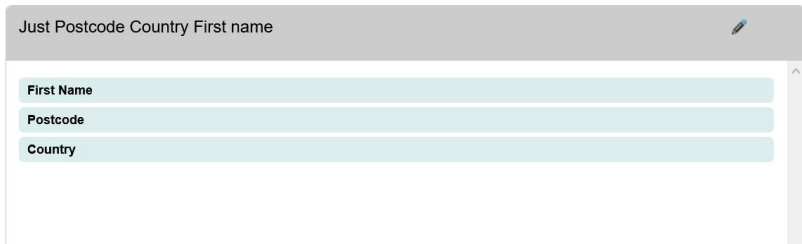
1st March 2021 ▼

The Export Group

- Way of specifying which supporter data fields we want to see e.g. first name, country, postcode.
- No need to expose email address, address 1, phone # ...etc

Account settings / Account data structure

Export Groups



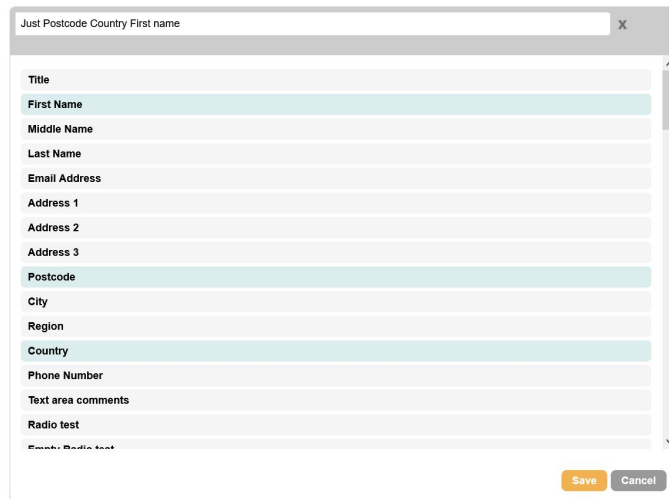
Just Postcode Country First name

First Name

Postcode

Country

[More on export groups](#)



Just Postcode Country First name X

Title

First Name

Middle Name

Last Name

Email Address

Address 1

Address 2

Address 3

Postcode

City

Region

Country

Phone Number

Text area comments

Radio test

Form: Radio test

Save Cancel

ENS API call #1: Authentication

POST

`https://e-activist.com/ens/service/authenticate`

Body: `12a3bc45-12ab-1a2b-9ab8-cde765f4gh32`

Response:

```
{
  ... "expires": 3600000,
  ... "ens-auth-token": "a56a56a56-b78b89-c34c34c34-de7878-9876aa5sx443"
}
```

[More details](#) on authenticating to ENS REST (~14 minutes in)

Also setting up an [authentication-only](#) server (~20 minutes in)

ENS API call #2: Profile data

GET

Headers:

```
Content-Type: application/json  
ens-auth-token: 1aa22a3c-c456-78910-a1b2-d3456ab78c90
```

URL:

```
https://politicalnetworks.com/ens/service/supporter/query?type=profile&daysBack=32&rows=100&profileId=1273&exportGroup=Just Postcode  
Country First name
```

Query parameters:

```
type: profile  
daysBack: 32 (maximum - actions from before this will not be returned)  
rows: 100 (maximum)  
start (optional): 1 (if more than 100 rows, specify the page)  
profileId: 1273  
exportGroup: Just Postcode Country First name (remember to urlEncode)
```

[More details](#)

ENS API call #2: Profile data

```
{
  "scores": [],
  "summary": {},
  "pagination": {
    "rows": 100,
    "total": 5,
    "start": 1
  },
  "data": [
    {
      "First Name": "Iain",
      "supporterId": 4863456,
      "Country": "RU",
      "Postcode": "127006",
      "createdOn": "2021-05-17"
    },
    {
      "modifiedOn": "2021-05-12",
      "First Name": "Iain ZA",
      "supporterId": 4863195,
      "Country": "ZA",
      "Postcode": "2190",
      "createdOn": "2021-05-11"
    }
  ]
}
```


Geocoding

- [Here](#), [Mapbox](#), [OpenCage](#)
- [Google geocoding](#)
- Set up a billing account
- Obtain an API key
- Set quotas

URL:

`https://maps.googleapis.com/maps/api/geocode/json?components=postal_code:1011|country:HU&key={apiKey}`

- `components=postal_code:1011|country:HU`
- Alternatively: `address=24%20Sussex%20Drive%20Ottawa%20ON`
 - (some countries don't use a postcode system)

Geocoding: response

```
{
  "results": [
    {
      "address_components": [
        {
          "long_name": "3",
          "short_name": "3",
          "types": [
            "street_number"
          ]
        },
        {
          "long_name": "Apor Péter utca",
          "short_name": "Apor Péter u.",
          "types": [
            "route"
          ]
        },
        {
          "long_name": "I. kerület",
          "short_name": "I. kerület",
          "types": [
            "political",
            "sublocality",
            "sublocality_level_1"
          ]
        },
        {
          "long_name": "Budapest",
          "short_name": "Budapest",
          "types": [
            "locality",
            "political"
          ]
        },
        {
          "long_name": "Hungary",
          "short_name": "HU",
          "types": [
            "country",
            "political"
          ]
        },
        {
          "long_name": "1011",
          "short_name": "1011",
          "types": [
            "postal_code"
          ]
        }
      ],
      "formatted_address": "Budapest, Apor Péter u. 3, 1011 Hungary",
      "geometry": {
        "location": {
          "lat": 47.4994905,
          "lng": 19.0392766
        },
        "location_type": "ROOFTOP",
        "viewport": {
          "northeast": {
            "lat": 47.5008394802915,
            "lng": 19.0406255802915
          },
          "southwest": {
            "lat": 47.4981415197085,
            "lng": 19.0379276197085
          }
        }
      },
      "place_id": "ChIJp4z9ED3cQUCvPR3cUDI_3I",
      "plus_code": {
        "compound_code": "F2XQ+QP Budapest, Hungary",
        "global_code": "8FVXF2XQ+QP"
      },
      "types": [
        "establishment",
        "lodging",
        "point_of_interest"
      ]
    },
    {
      "status": "OK"
    }
  ]
}
```

The Server

Geocoding and adding a supporter

The Server

▼ 2:

latitude: "48.8640493"

longitude: "2.3310526"

firstName: "Iain FR"

▼ 3:

latitude: "37.9718603"

longitude: "23.7304751"

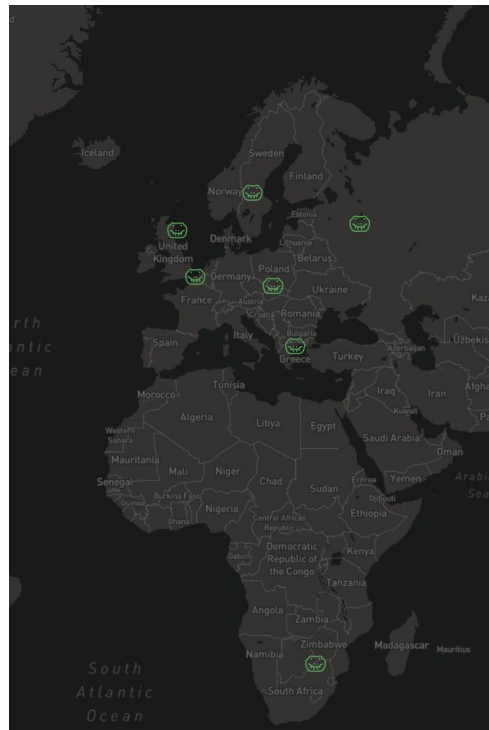
firstName: "Iain Greece"

▼ 4:

latitude: "-26.2573212"

longitude: "28.0478826"

firstName: "Iain ZA"



Implementation

- Code available on [github](#)
- Keep your API keys - **particularly your ENS private key** - safe at all times
- Processing supporter data on a server outside EN
 - Make sure it's secure e.g. require HTTPS, check domains
- Displaying supporter data
 - Get relevant permissions from supporters e.g. via opt-in
- “Secure” and “relevant permissions” as defined by your policies and relevant data laws
- This code is for demo purposes
 - Not been security tested
 - Not informed by particular data protection advice
 - Meant as a starting point or prototype

Thank You.

Questions and answers

iaian@engagingnetworks.net