

# Ajax 2: FourSquare API in Node

---

# Agenda

---

- Foursquare Service
- FourSquare Developer
- Checkin API
- Insomnia REST Client
- Node.js program to access API, and retrieve checkin lists for a location

# FOURSQUARE

Find the best places to eat, drink, shop, or visit in any city in the world.

Access over 75 million short tips from local experts.

I'm looking for...

Waterford

 Search



Food



Coffee



Nightlife



Fun



Shopping

## Suggestions for Best Nearby near Waterford

Filters: Specials Haven't Been Following Price Open Now Saved Liked

## 1. Bodega!

8.9 54 John St, Waterford Mediterranean • €€€

Lewis T. • December 2, 2015

Came here with a company night out (40 people). Great service from the staff, tasty food and wide menu. I didn't have dessert but the brownie with jelly tots looked (and apparently tasted) fabulous!



Save

## 2. Geoff's Café Bar

8.4 7-9 John St, Waterford Pub • €€€ • View Menu

Sandra B. • July 3, 2014

Great food and nice people. We had fish and chips, the cheeseburger, and the goat cheese crostini. Each was better than the last.



Save

## 3. Emilianos

8.2 21 High St., Waterford Italian • €€€

jabit • July 8, 2010

This is a really nice restaurant, pizza and pasta are fab but I'm guessing everything on the menu is gorgeous. Lovely staff and atmosphere too.



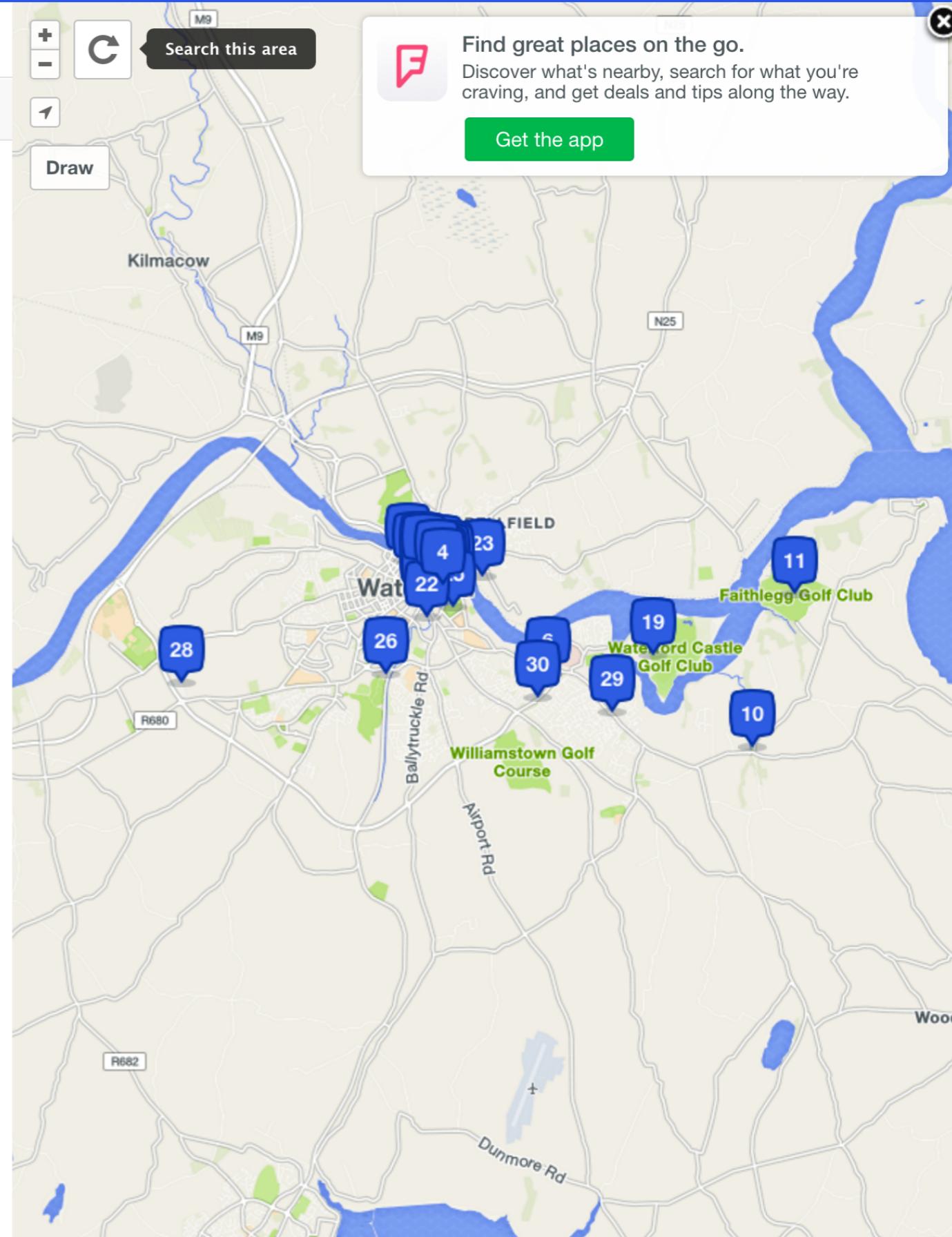
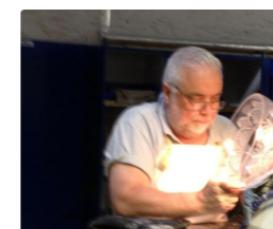
Save

## 4. House of Waterford Crystal

8.1 The Mall, Waterford Museum

Hotelsireland • March 9, 2011

Perfect location in the centre of town. Very interesting. You can actually see live how the crystal is made and talk to the artists there.



Top Picks

Trending

Food

Coffee

Nightlife

Fun

Shopping



## Find great places on the go.

Discover what's nearby, search for what you're craving, and get deals and tips along the way.

[Get the app](#)

## My Apps

**test****Client ID**

ZNRM2 [REDACTED] BFSRXMBWXT2

**Client Secret**

HZPPG [REDACTED] OHBY0LXX01CC

[More about this app...](#)[Create a new app](#)**Access Token URL**[https://foursquare.com/oauth2/access\\_token](https://foursquare.com/oauth2/access_token)**Authorize URL**<https://foursquare.com/oauth2/authorize>**Learn more about:**

- [OAuth2 and foursquare](#)
- [The foursquare API](#)

# Credentials in Javascript

---

End Point of Interest

```
var fsConfig = {  
  base_url: 'https://api.foursquare.com/v2/venues/explore?',  
  client_id: 'Your ID',  
  client_secret: 'Your Secret',  
};  
  
console.log (fsConfig);
```

App Keys

# API Documentation

## Users

## Venues

search  
suggestcompletion  
categories  
timeseries  
trending  
explore  
add  
managed  
**Aspects**  
similar  
photos  
events  
likes  
nextvenues  
hours  
stats  
links  
menu  
tips  
herenow  
listed  
**Actions**  
claim  
dislike  
flag  
proposeedit  
like  
setrole  
edit  
setsinglelocation

## Venuegroups

## Checkins

## Tips

## Lists

## Updates

# Explore Recommended and Popular Venues

<https://api.foursquare.com/v2/venues/explore>

Returns a list of recommended venues near the current location.

If authenticated, the method will potentially personalize the ranking based on you and your friends. If you do not authenticate, you will not get this personalization.

This endpoint is part of the [venues API](#).

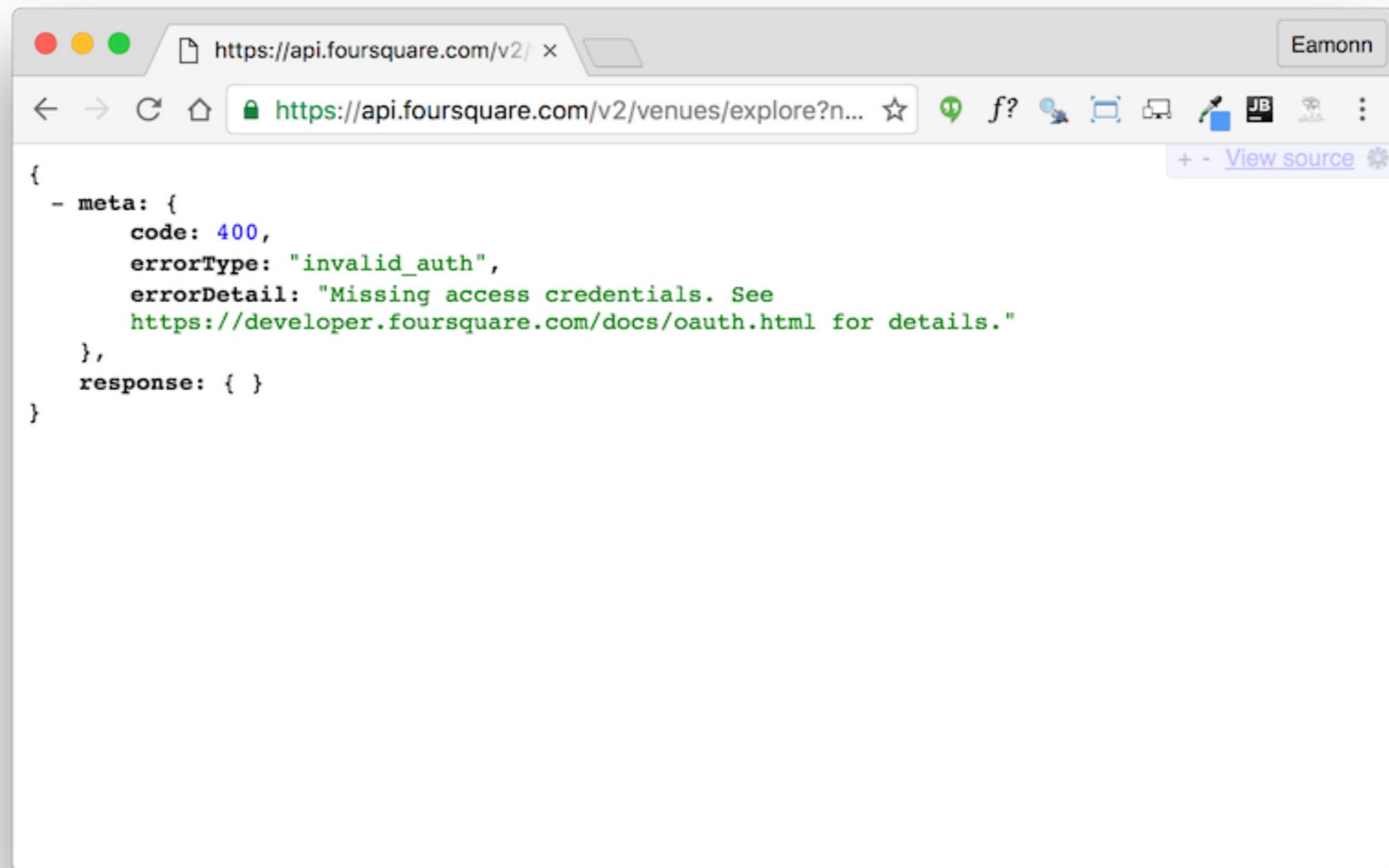
<b>HTTP Method</b>	GET
<b>Requires Acting User</b>	No ( <a href="#">learn more</a> )
<b>Modes supported</b>	<b>foursquare</b> ( <a href="#">learn more</a> )

## Parameters

All parameters are optional, unless otherwise indicated.

<b>ll</b>	44.3,37.2	<b>required</b> unless <b>near</b> is provided. Latitude and longitude of the user's location.
<b>near</b>	Chicago, IL	<b>required</b> unless <b>ll</b> is provided. A string naming a place in the world. If the <b>near</b> string is not geocodable, returns a <b>failed_geocode</b> error. Otherwise, searches within the bounds of the geocode and adds a <b>geocode</b> object to the response.
<b>llAcc</b>	10000.0	Accuracy of latitude and longitude, in meters.
<b>alt</b>	0	Altitude of the user's location, in meters.
<b>altAcc</b>	10000.0	Accuracy of the user's altitude, in meters.
<b>radius</b>	250	Radius to search within, in meters. If radius is not specified, a suggested radius will be used based on the density of venues in the area.
<b>section</b>	food	One of <b>food</b> , <b>drinks</b> , <b>coffee</b> , <b>shops</b> , <b>arts</b> , <b>outdoors</b> , <b>sights</b> , <b>trending</b> or <b>specials</b> , <b>nextVenues</b> (venues frequently visited after a given venue), or <b>topPicks</b> (a mix of recommendations generated without a query from the user).

<https://api.foursquare.com/v2/venues/explore?near=Waterford,IE&query=&v=20140601>



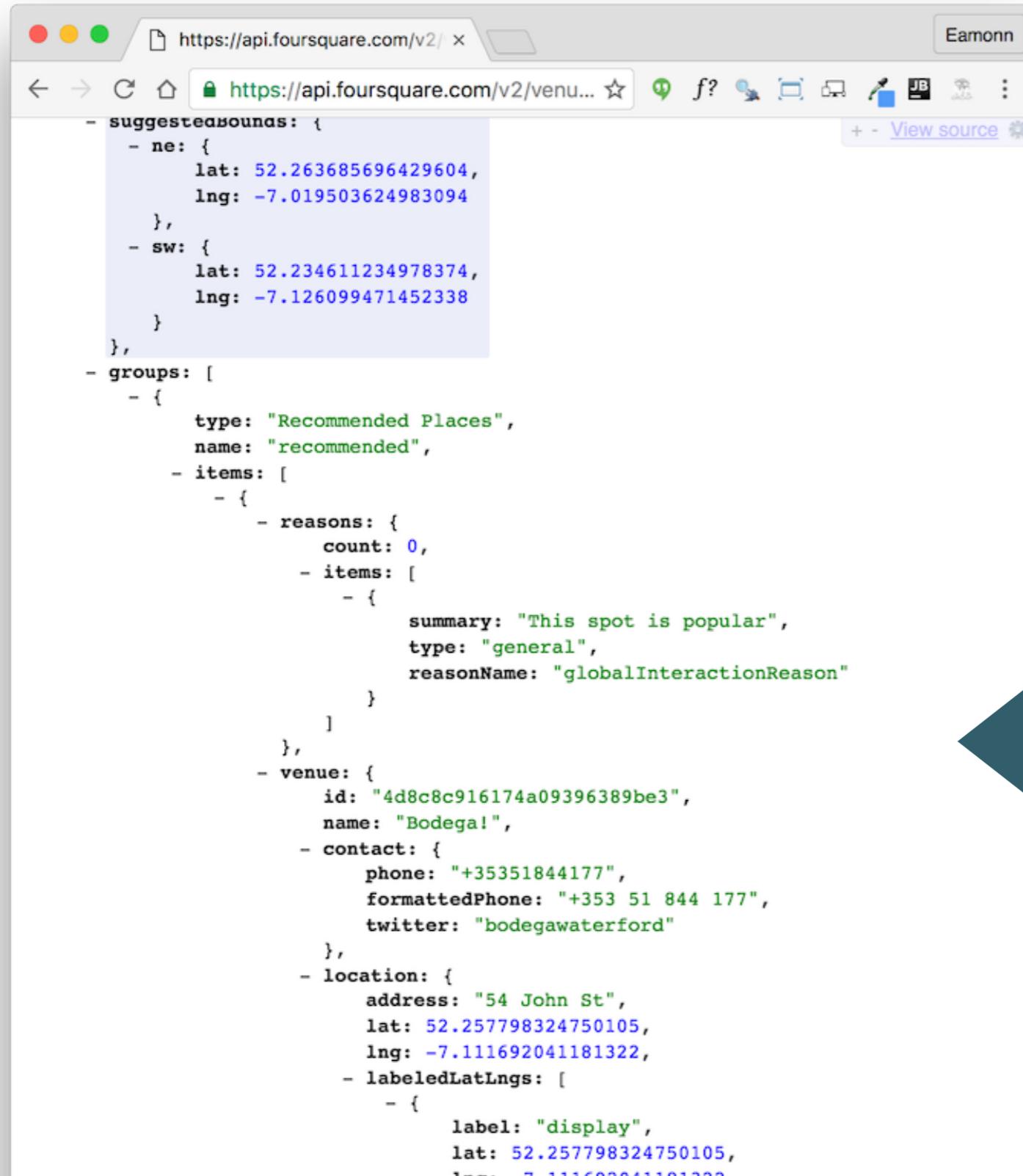
The screenshot shows a web browser window with the following details:

- Address Bar:** https://api.foursquare.com/v2/venues/explore?near=Waterford,IE&query=&v=20140601
- User:** Eamonn
- Toolbar:** Includes standard browser icons for back, forward, search, and refresh.
- Content Area:** Displays a JSON error response. The response is as follows:

```
{  
  - meta: {  
    code: 400,  
    errorType: "invalid_auth",  
    errorDetail: "Missing access credentials. See  
    https://developer.foursquare.com/docs/oauth.html for details."  
  },  
  response: {}  
}
```

# GET Request

`https://api.foursquare.com/v2/venues/explore?near=Waterford,IE&query=&client_id=YOUR_ID&client_secret=YOUR_SECRET&v=201`



```
https://api.foursquare.com/v2/venues/explore?near=Waterford,IE&query=&client_id=YOUR_ID&client_secret=YOUR_SECRET&v=201

{
  "suggestions": [
    {
      "ne": {
        "lat": 52.263685696429604,
        "lng": -7.019503624983094
      },
      "sw": {
        "lat": 52.234611234978374,
        "lng": -7.126099471452338
      }
    }
  ],
  "groups": [
    {
      "type": "Recommended Places",
      "name": "recommended",
      "items": [
        {
          "reasons": {
            "count": 0,
            "items": [
              {
                "summary": "This spot is popular",
                "type": "general",
                "reasonName": "globalInteractionReason"
              }
            ]
          }
        }
      ],
      "venue": {
        "id": "4d8c8c916174a09396389be3",
        "name": "Bodega!",
        "contact": {
          "phone": "+35351844177",
          "formattedPhone": "+353 51 844 177",
          "twitter": "bodegawaterford"
        },
        "location": {
          "address": "54 John St",
          "lat": 52.257798324750105,
          "lng": -7.111692041181322,
          "labeledLatLngs": [
            {
              "label": "display",
              "lat": 52.257798324750105,
              "lng": -7.111692041181322
            }
          ]
        }
      }
    }
  ]
}
```

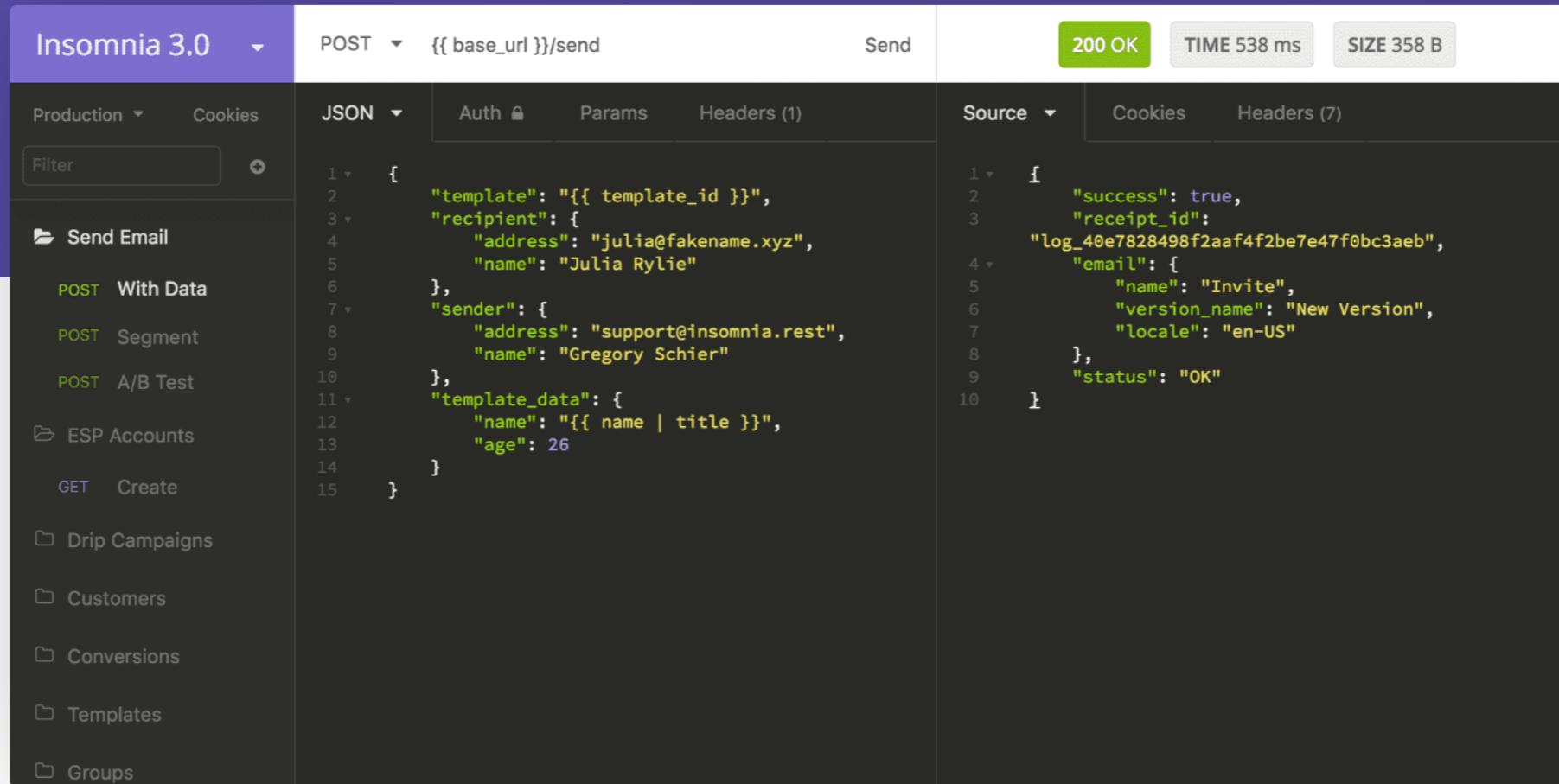
Your Credentials

JSON response

# A simple and beautiful REST API client

Mac, Windows, and Linux

[Download Free for Mac](#)



The screenshot shows the Insomnia 3.0 interface with a POST request to `{{ base_url }}/send`. The request body is JSON, containing a template with a recipient and sender, and a template data object. The response is a 200 OK status with a success message, receipt ID, log ID, email details, and a status of "OK".

Request URL: `POST {{ base_url }}/send`

Request Headers: `Content-Type: application/json`

Request Body (JSON):

```
1  {
2    "template": "{{ template_id }}",
3    "recipient": {
4      "address": "julia@fakename.xyz",
5      "name": "Julia Rylie"
6    },
7    "sender": {
8      "address": "support@insomnia.rest",
9      "name": "Gregory Schier"
10   },
11   "template_data": {
12     "name": "{{ name | title }}",
13     "age": 26
14   }
15 }
```

Response Headers: `200 OK`, `TIME 538 ms`, `SIZE 358 B`

Response Body (JSON):

```
1  {
2    "success": true,
3    "receipt_id": "log_40e7828498f2aaaf4f2be7e47f0bc3aeb",
4    "email": {
5      "name": "Invite",
6      "version_name": "New Version",
7      "locale": "en-US"
8    },
9    "status": "OK"
10 }
```

GET https://api.foursquare.com/v2/venues/explore?near= Send

400 Bad Request

TIME 190 ms

SIZE 495 B

Insomnia

No Environment

Cookies

Filter



GET explore-no-auth

GET explore-auth

JSON

Auth

Query

Headers (1)

URL PREVIEW

```
https://api.foursquare.com/v2/venues/explore?near=Waterford,  
IE&query=&v=20140601
```

name

value

Source

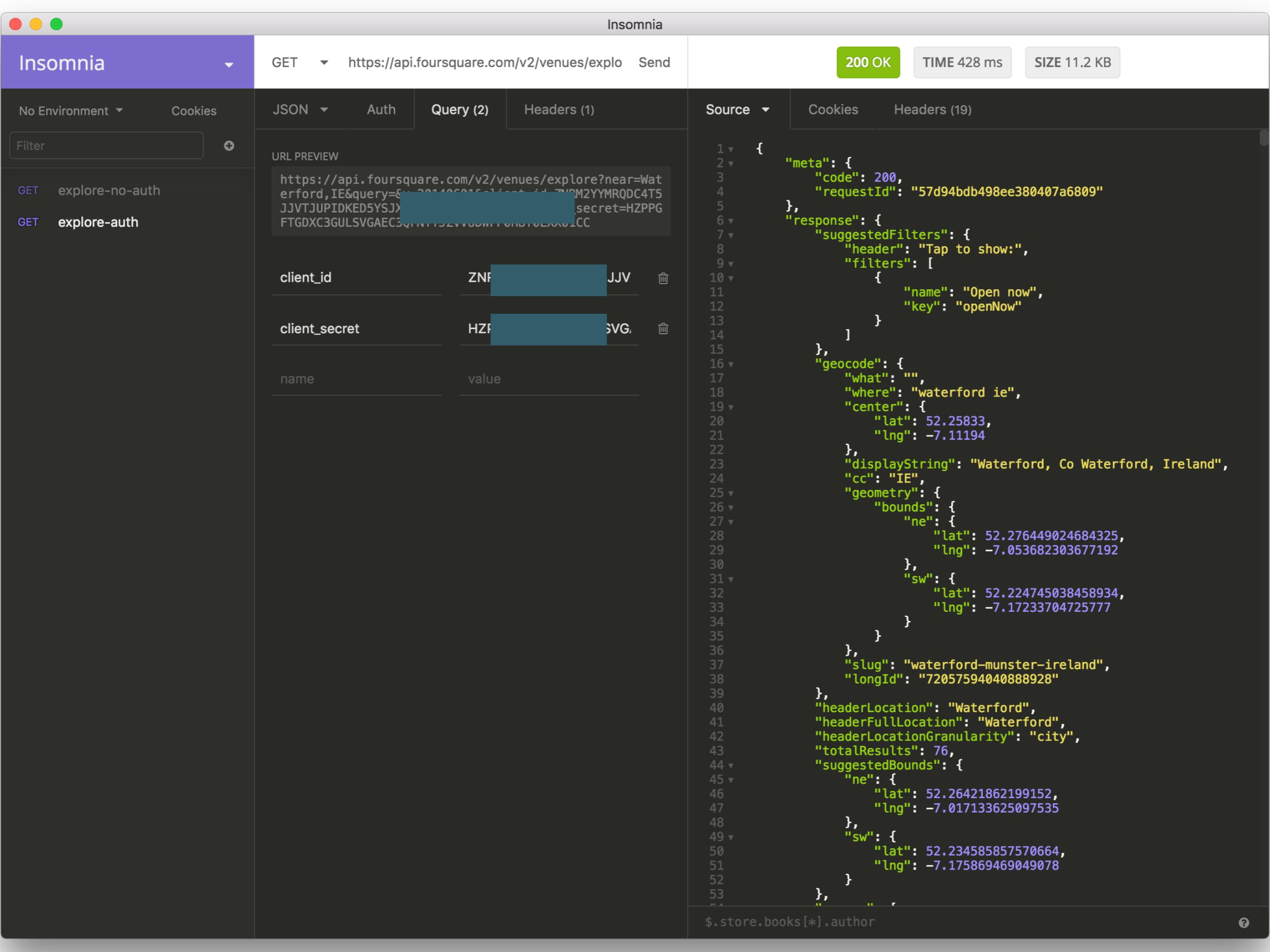
Cookies

Headers (12)

```
1 {  
2   "meta": {  
3     "code": 400,  
4     "errorType": "invalid_auth",  
5     "errorDetail": "Missing access credentials. See  
6     https://developer.foursquare.com/docs/oauth.html for details."  
7   },  
8   "response": {}  
}
```

\$.store.books[\*].author





- 3,374 ‘Lines’ of JSON!

```
40601&cli
2YYMRQDC4T
ED5YSJX3FW
T2&client_
GFTGDXC3GU
TJZVVUDWF
CC

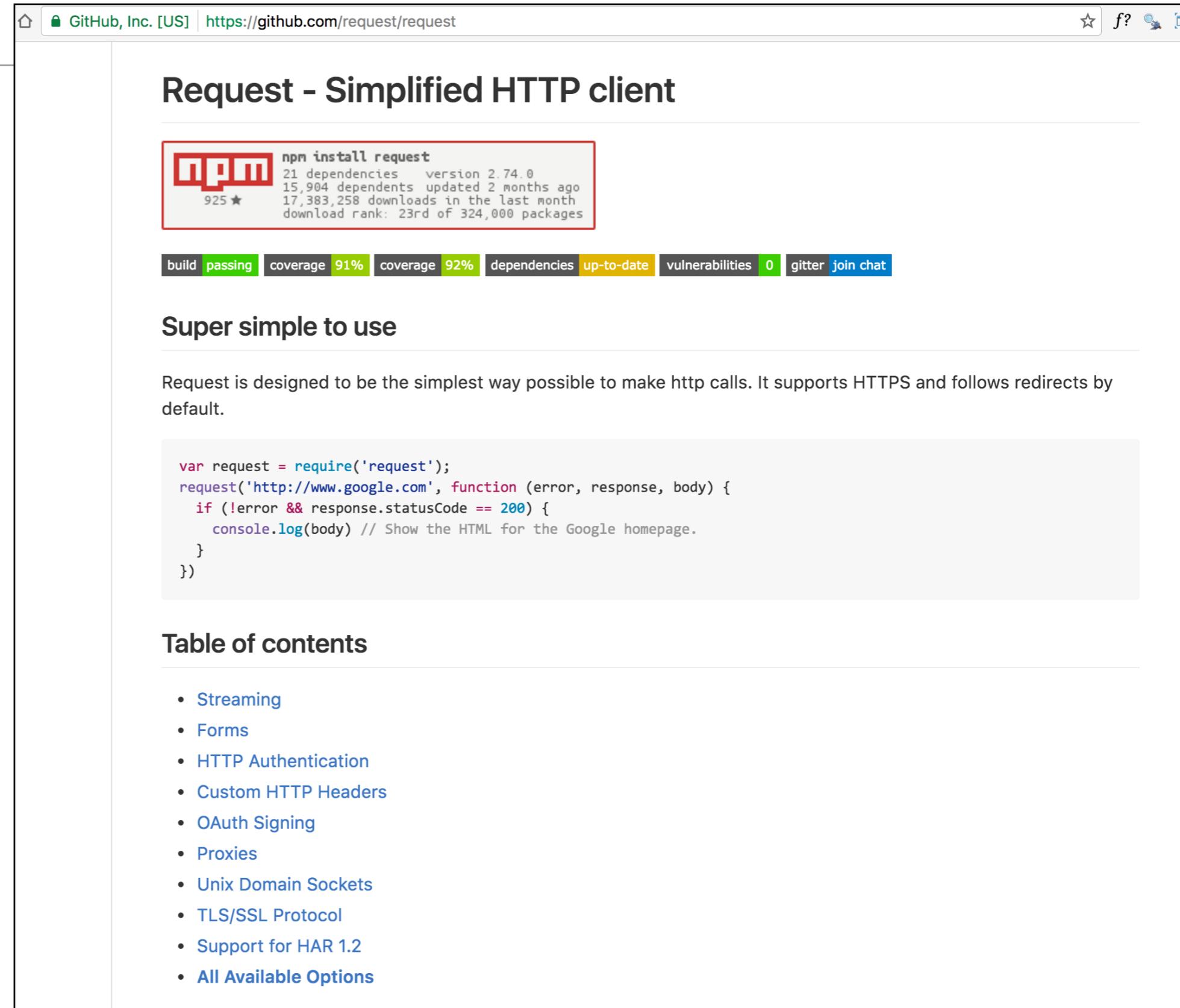
NRM2' 3330
3331
3332
3333 ▾
3334
3335
3336
3337 ▾
3338
3339
3340
3341
3342
3343 ▾
3344 ▾
3345
3346
3347
3348
3349
3350
3351
3352
3353 ▾
3354
3355
3356 ▾
3357
3358
3359
3360
3361 ▾
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374

"rating": 6.5,
"ratingSignals": 5,
"allowMenuUrlEdit": true,
"photos": {
  "count": 0,
  "groups": []
},
"hereNow": {
  "count": 0,
  "summary": "Nobody here",
  "groups": []
}
},
"tips": [
{
  "id": "54047319498e008a39442af4",
  "createdAt": 1409577753,
  "text": "Plenty of parking",
  "type": "user",
  "canonicalUrl": "https://foursquare.com/item/54047319498e008a39442af4",
  "logView": true,
  "agreeCount": 0,
  "disagreeCount": 0,
  "todo": {
    "count": 0
  },
  "user": {
    "id": "2463172",
    "firstName": "Tony",
    "lastName": "Costeloe",
    "gender": "male",
    "photo": {
      "prefix": "https://irs1.4sqi",
      "suffix": "/BASRNTVX0GB5QHTN"
    }
  }
},
],
"referralId": "e-0-53d6768f498efde7e4ba23da-2
}
```

# Api Programming

npm install request

- Node Module simplifying interacting with HTTP in JavaScript



The screenshot shows the GitHub page for the 'request' npm module. The page title is 'Request - Simplified HTTP client'. It features an 'npm' logo with a red border, showing statistics: 21 dependencies, version 2.74.0, 15,904 dependents updated 2 months ago, 17,383,258 downloads in the last month, and a download rank of 23rd of 324,000 packages. Below the logo are buttons for 'build' (passing), 'coverage' (91%), 'dependencies' (up-to-date), 'vulnerabilities' (0), 'gitter', and 'join chat'. A section titled 'Super simple to use' contains sample code for making an HTTP request to Google's homepage. The code is as follows:

```
var request = require('request');
request('http://www.google.com', function (error, response, body) {
  if (!error && response.statusCode == 200) {
    console.log(body) // Show the HTML for the Google homepage.
  }
})
```

Below the code is a 'Table of contents' section with links to various features:

- Streaming
- Forms
- HTTP Authentication
- Custom HTTP Headers
- OAuth Signing
- Proxies
- Unix Domain Sockets
- TLS/SSL Protocol
- Support for HAR 1.2
- All Available Options

# Node Api Request

```
const request = require('request');
const fsConfig = require('./fs-config');

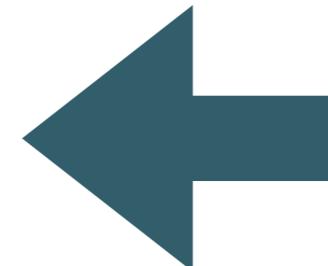
var fsCredentials = '&client_id=' + fsConfig.client_id + '&client_secret='
                    + fsConfig.client_secret + '&v=20140601';

function loadVenues(locationName, venueKeyword) {

  var requestOptions = {
    url: fsConfig.base_url + 'near=' + locationName + '&query=' + venueKeyword + fsCredentials,
    method: 'GET',
    json: {},
  };

  request(requestOptions, (err, response, body) => {
    const venues = body.response.groups[0].items;
    console.log(venues);
  });
}

var locationName = 'Waterford, IE';
loadVenues(locationName, '');
```



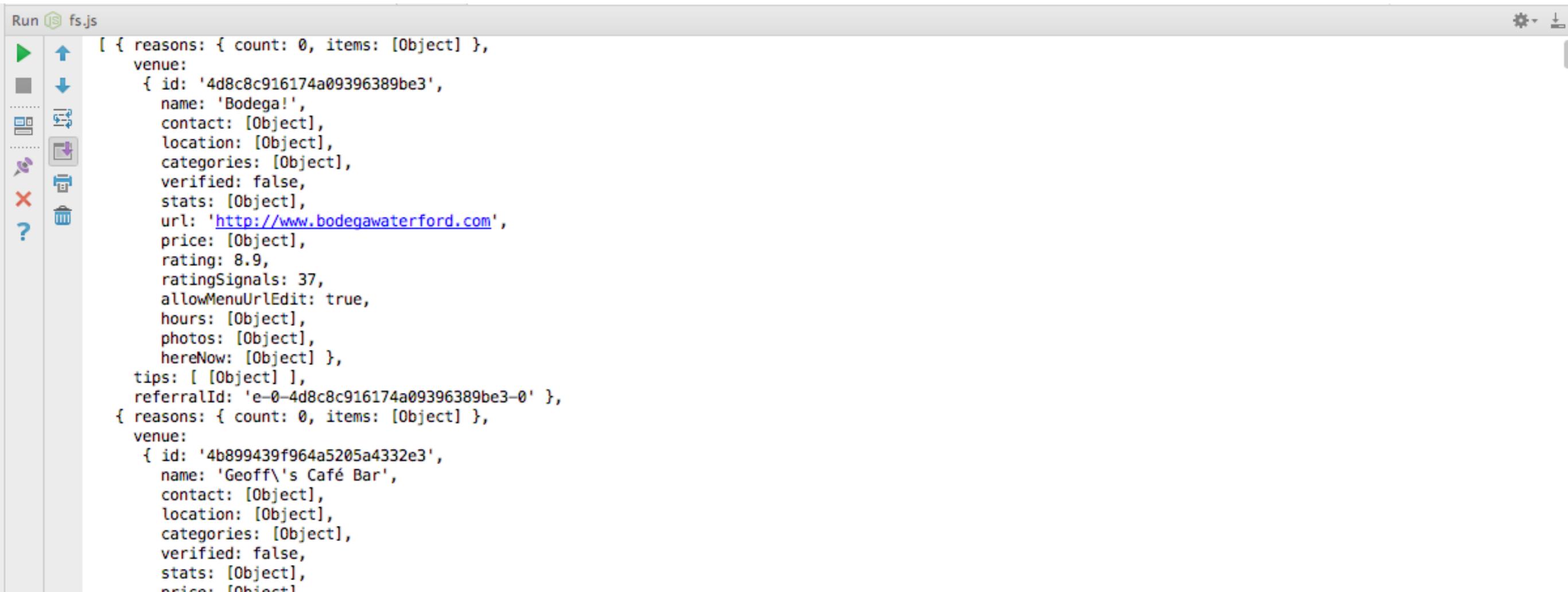
Callback invoked  
when response  
received

```
request(requestOptions, (err, response, body) => {

  const venues = body.response.groups[0].items;
  console.log(venues);

});
```

Run fs.js



```
[ { reasons: { count: 0, items: [Object] },
  venue:
  { id: '4d8c8c916174a09396389be3',
    name: 'Bodega!',
    contact: [Object],
    location: [Object],
    categories: [Object],
    verified: false,
    stats: [Object],
    url: 'http://www.bodegawaterford.com',
    price: [Object],
    rating: 8.9,
    ratingSignals: 37,
    allowMenuUrlEdit: true,
    hours: [Object],
    photos: [Object],
    hereNow: [Object] },
  tips: [ [Object] ],
  referralId: 'e-0-4d8c8c916174a09396389be3-0' },
  { reasons: { count: 0, items: [Object] },
  venue:
  { id: '4b899439f964a5205a4332e3',
    name: 'Geoff\'s Café Bar',
    contact: [Object],
    location: [Object],
    categories: [Object],
    verified: false,
    stats: [Object],
    price: [Object] } }
```

```
request(requestOptions, (err, response, body) => {  
  
  const venues = body.response.groups[0].items;  
  console.log(venues);  
  
});
```



Declare local array  
Loop through venues  
create 'checkin' object  
store in array  
print array

```
request(requestOptions, (err, response, body) => {  
  
  const venues = body.response.groups[0].items;  
  const checkins = [];  
  
  for (let venue of venues) {  
    const checkin = {  
      name: venue.venue.name,  
      checkins: venue.venue.stats.checkinsCount,  
      users: venue.venue.stats.usersCount,  
    };  
    checkins.push(checkin);  
  }  
  
  console.log(checkins);  
});
```

```
request(requestOptions, (err, response, body) => {

  const venues = body.response.groups[0].items;
  const checkins = [];

  for (let venue of venues) {
    const checkin = {
      name: venue.venue.name,
      checkins: venue.venue.stats.checkinsCount,
      users: venue.venue.stats.usersCount,
    };
    checkins.push(checkin);
  }

  console.log(checkins);
});
```

```
[ { name: 'Bodega!', checkins: 181, users: 141 },
  { name: 'Geoff\'s Café Bar', checkins: 808, users: 286 },
  { name: 'Emilianos', checkins: 71, users: 57 },
  { name: 'House of Waterford Crystal', checkins: 941, users: 737 },
  { name: 'Carter\'s Chocolate Cafe', checkins: 159, users: 55 },
  { name: 'Ardkeen Quality Food Store', checkins: 585, users: 106 },
  { name: 'The Gingerman', checkins: 166, users: 129 },
  { name: 'Tower Hotel & Leisure Centre',
    checkins: 787,
    users: 248 },
  { name: 'McLeary\'s', checkins: 61, users: 50 },
  { name: 'Jack Meades', checkins: 96, users: 74 },
  { name: 'Faithlegg House Hotel', checkins: 149, users: 101 },
  { name: 'Café Goa', checkins: 178, users: 57 },
  { name: 'The Reg', checkins: 288, users: 136 },
  { name: 'Theatre Royal', checkins: 217, users: 81 },
  { name: 'The Book Centre', checkins: 332, users: 131 },
  { name: 'The Munster Bar', checkins: 179, users: 127 },
  { name: 'La Bohème', checkins: 35, users: 32 },
  { name: 'Medieval Museum', checkins: 135, users: 112 },
  { name: 'Waterford Castle Hotel & Golf Resort',
    checkins: 335,
    users: 219 },
  { name: 'No.9 Café', checkins: 83, users: 39 },
  { name: 'Reginald\'s Tower', checkins: 444, users: 181 },
  { name: 'Costa Coffee', checkins: 123, users: 44 },
  { name: 'Athenaeum House Hotel', checkins: 395, users: 55 },
  { name: 'Bishop\'s Palace', checkins: 162, users: 116 },
  { name: 'The Donal Lio's Park', checkins: 400, users: 127 } ]
```