# Find Printers Node JS Server Documentation

**Current URL:** http://fpserver.herokuapp.com

## Connecting

All services and functionalities of the server are available simply by connecting to the server's URL using a Socket.IO library. A socket connection will be established, and will last for the entire session. The server's URL can also be loaded in a browser or WebView, as there's a build-in HTML based client. However, for mobile apps the performance will most likely be better by using only the Socket.IO connection along with whatever Google Maps API the platform can use natively.

## Handles

### Requests

| **Emit message (to server)** | **Parameters** | **Description** |
|---|---|---|
| *connection* | *(none)* | *Main connection event, emitted automatically as the connection is made. Will return a unique session ID, which must be stored and used for all other emissions.* |
| getCount | bounds (bounds object*) | The server will count all markers within the provided bounds. |
| getLibraries | bounds (bounds object*) | The server will get all markers within the given bounds. |
| getClusters | bounds (bounds object*),<br>zoom (int) | The server will create clusters of locations within the given bounds. |
| stopEverything | (none) | Will stop all active queries for the session. It's a good idea to call this first, every time the client needs a new set of markers from the server. |

* Structure of bounds object:

```
{
    latA: (smallest lat, float),
    latB: (biggest lat, float),
    lngA: (smallest lng, float),
    lngB: (biggest lng, float),
    gridsize: (amount of map subdivisions, int, 2 works best)
}
```

# Responses

| **Emit message (from server)** | **Return data** | **Description** |
|---|---|---|
| *connect* | *(none)* | *When the first connection has been made. It's a good idea to place the other socket listeners inside this one.* |
| countLibraries | rows (Array of objects) | Only one index is in the array, and this index' only child object has one parameter: **"count" (int)**. Use this number to determine if the app should get markers or get clusters. |
| getLibraries | rows (Array of objects) | Each index has an object. Each object has<br>**'name' (String)**,<br>**'lat' (float)** and<br>**'lng' (float)** properties. |
| getCluster | rows (Array of objects),<br>isLast (boolean) | One emission will be made for each subdivided grid inside the provided bounds. Each index has an object with properties<br>**'lat' (float)**,<br>**'lng' (float)**,<br>**'count' (int)** and<br>'**country' (int)**.<br>Used to create fake clusters (simply create a marker for each index, using the cluster texture and setting 'count' as the marker's label).<br>'country' can be used for additional filtering on the client. |
| removeMarkers | (none) | The server will sometimes ask the client to remove all current markers, with this message, so tie this listener to a function that removes all markers and clusters and everything on the map view. |
| serverError | error (Object) | Used by server to inform client that an error occurred. The error object has **'code'** and **'status'** properties (both strings). |