

# PolitiGo Backend README

## Project Overview

PolitiGo is a NYC Political event lister SPA, in an effort to activate users and connect them to one another based on shared political causes. Users can add to their attending list from all events listed, and can view relevant details including the mapped location pinpoint. As a fun incentive, a user has a “humanitarian goodwill score” that starts out at 0, and is increased by 10 every time they add an event and decreased by 10 every time they remove an event. To help foster a sense of community and keep attendance high, users can also view other users’ scores (and event list). 🦊

## Technologies Used

JavaScript, Ruby on Rails, Google Places API, HTML, CSS, JSON, Materialize CDN & GoogleFonts, ActiveModel Serializers, Ruby gems

## Getting Started

### Prerequisites

To use, clone down this repo and open with your preferred text editor. This project uses Ruby on Rails, so you’ll need to make sure you have both installed (if you don’t, visit [this link for instructions: Digital Ocean: Install Ruby and Set Up Local Environment](#)) as well as PostgreSQL as the database resource.

### Installing

The project gemfile specifies Ruby 2.6.0, but was created with 2.5.3 so make sure you have Ruby 2.5.3 or higher. Open terminal and in the root of the project directory run:

```
bundle install  
or simply  
bundle
```

to install gem dependencies. Once that completes successfully, run:

```
rake db: create && rake db:migrate
```

to facilitate backend set up. You can then:

```
rails c
```

and jump into the console to test out that tables and relationships were established correctly, that class instances can be created and related successfully, etc. For events, this project uses seed data, with the future intention of utilizing NYC official city

government data. Until then, you can use pre-existing seed data or make your own, but either way, run:

```
rails db:seed
```

to create a starter list of events. If the development database is running correctly, then the seed data should be successfully created and final step is to test the server. To activate it, run:

```
rails s
```

and once the terminal says it's running, navigate to <http://localhost:3000>. If there's "Yay! You're on Rails!" welcome page, then you can visit <http://localhost:3000/api/v1/users> or [api/v1/events](http://localhost:3000/api/v1/events) or [api/v1/user\\_events](http://localhost:3000/api/v1/user_events) to check that all the data is being rendered properly. If so, it's time for the frontend (link below). 👍

## Frontend Link

[politigo\\_frontend](#)

## Demo Video

[politigo.mov](#)

## Authors

**HAMILTON HUDSON**

» [ehamiltonhudson@gmail.com](mailto:ehamiltonhudson@gmail.com)

↳ *LinkedIn*: <https://www.linkedin.com/in/hamiltonhudson>

↳ *Website*: <https://hamiltonhudson.myportfolio.com>

↳ *Blog*: <https://ehhudson.wordpress.com>

↳ *Twitter*: <https://twitter.com/HamiltonHudson>

## License

This project is licensed under the MIT License - see the LICENSE.md file for details.