

groupie-tracker

Objectives

Groupie Trackers consists on receiving a given API and manipulate the data contained in it, in order to create a site, displaying the information.

- It will be given an [API](#), that consists in four parts:
 - The first one, `artists`, containing information about some bands and artists like their name(s), image, in which year they began their activity, the date of their first album and the members.
 - The second one, `locations`, consists in their last and/or upcoming concert locations.
 - The third one, `dates`, consists in their last and/or upcoming concert dates.
 - And the last one, `relation`, does the link between all the other parts, `artists`, `dates` and `locations`.
- Given all this you should build a user friendly website where you can display the bands info through several data visualizations (examples : blocks, cards, tables, list, pages, graphics, etc). It is up to you to decide how you will display it.
- This project also focuses on the creation of events/actions and on their visualization.
 - The event/action we want you to do is known as a client call to the server (client-server). We can say it is a feature of your choice that needs to trigger an action. This action must communicate with the server in order to receive information, ([request-response])(<https://en.wikipedia.org/wiki/Request%E2%80%93response>)
 - An event consists in a system that responds to some kind of action triggered by the client, time, or any other factor.

Instructions

- The backend must be written in **Go**.
- The site and server cannot crash at any time.
- All pages must work correctly and you must take care of any errors.
- The code must respect the [good practices](#).
- It is recommended to have **test files** for [unit testing](#).

Allowed packages

- Only the [standard Go](#) packages are allowed.

Usage

- You can see an example of a RESTful API [here](#)

This project will help you learn about :

- Manipulation and storage of data.
- [JSON](#) files and format.
- HTML.
- Event creation and display.
- [Client-server](#).