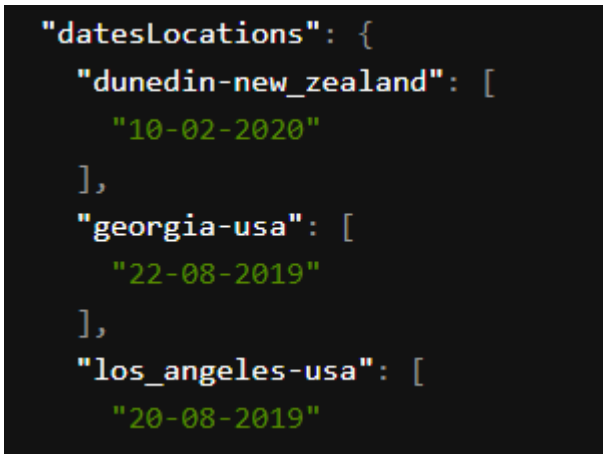
🎵 Groupie Tracker

Relations

Vous l'avez peut-être remarqué, mais la partie `relation` de l'API est un peu différente des autres, elle contient ces données :



Les clés étant dynamiques, on ne peut pas les utiliser de manière standard en JSON. On va donc utiliser un autre procédé.

Admettons que nous ne connaissons pas le nom des clés, nous allons utiliser un type *map* pour notre `datesLocations` :

```
type Date struct {
  Id int `json:"id"`
  DatesLocations map[string][]string `json:"datesLocations"`
}
```

Qu'est-ce qu'on fait ici ? Basiquement, chaque endroit est "relié" à un tableau de string qui correspond à une date, soit pour chaque string > []string. Ne connaissant pas le nom de la clé initiale, on dit à DatesLocations de "mapper" chaque string qu'il trouve, donc par exemple `dunedin-new_zealand` à un tableau de string lui appartenant.

Variables Go vers Javascript

Il est possible de faire passer ses variables Go en Javascript, c'est un peu tricky mais ça marche. Vous avez besoin de faire ça pour effectuer la barre de recherche et réussir le bonus `geolocalization`. Pour se faire, commencez par alimenter votre fichier .js comme ceci

```
let tableauAvecTousLesArtistes = []
let unSeulArtiste = ""
```

Puis, selon ce que vous voulez faire, modifiez votre HTML comme suit. Si vous souhaitez ajouter tous les artistes à une array en Javascript :

```
{{ range .Artists }}
<!-- Votre code -->
<script>
    tableauAvecTousLesArtistes.push("{{ . }}")
</script>
{{ end }}
```

Si vous voulez uniquement passer une seule valeur, en backend Go, ça restera classique, il faudra juste vérifier que vous ayez bien la bonne valeur passée dans le template exécuté :

```
package groupie

type Artist struct {
  Name string `json:"name"`
}

data := Artist{
  Name: "Micheal Jackson"
}

tmpl.ExecuteTemplate(w, "layout", Artist)
```

```
<script>
    unSeulArtiste = "{{ .Name }}"
    console.log(unSeulArtiste) // Vous devriez avoir Micheal Jackson d'affiché
</script>
```

Une fois ceci fait, vous avez la possibilité de tout coder dans la balise script, ce qui est très moche et déconseillé, ou alors vous avez la possibilité de renseigner un fichier .js pour rentrer votre code :

```
<script src="static/script.js"></script>
```

Tada ! La/les variable(s) que vous avez passé devrai(en)t se retrouver ici.

Range clé - valeur

Pour afficher en HTML au choix votre clé, votre valeur, ou les deux, venant d’une map, la syntaxe est simple :

```
{{ range $key, $value := .VotreMap }}
   <p>{{ $key }}</p>: <p>{{ $value }}</p>
{{ end }}
```