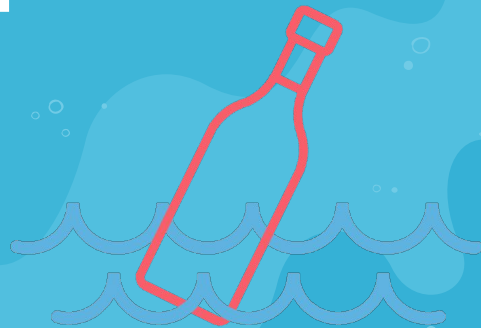


Bouteille à la mer

Laoun N. – Wyss V. – Marty M.



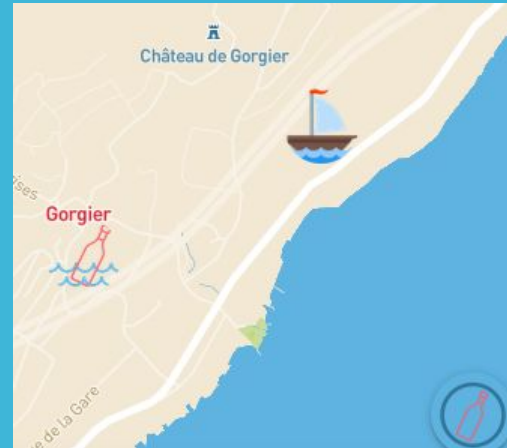
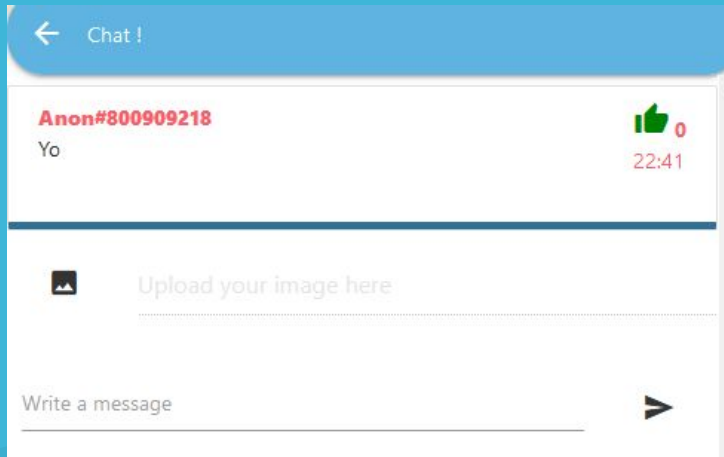
Sommaire

1. Introduction
2. Démonstration
3. Websocket
4. Queue
5. Gestion des id
6. Mapbox
7. Vue.js
8. Materialize et sass
9. Conclusion



1. Introduction

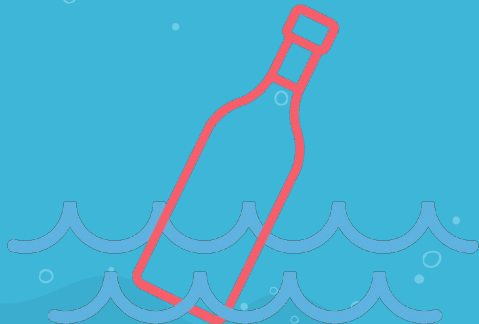
- Messagerie instantanée
- Géolocalisée et temporaire
- Anonyme



2. Démonstration

3. Websockets

1. Pull et Push entre le client et le serveur rapide
2. Pratique pour envoyer et recevoir des messages
3. Nécessite l'exécution d'un nouveau processus
 - a. `php artisan websockets:init`
4. Modification du `nginx.conf` pour créer un proxy wss
5. Récupération de la session du client grâce à ses cookies



4. Queue

- Delay un job

```
DeleteConversation::dispatch($conv)->delay($timeOfDeath);
```

- Évite de tester tous les x temps
- Nécessite quelques commande supplémentaires

```
php artisan queue:table
```

```
php artisan queue:work
```

```
php artisan make:job DeleteConversation
```

```
php artisan queue:listen
```

5. Gestion des id

- L'id d'un non connecté est random et négatif
- L'id est utilisé comme nom des messages

LaMousseAuLini

asd

Anon#61604

vf

Nicolas

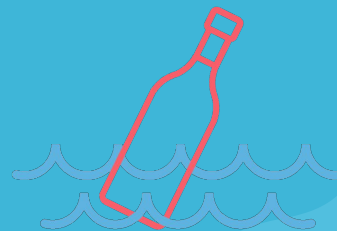
f



6. Mapbox



1. Bibliothèque pour afficher des maps
2. Facile d'utilisation
3. Editeur de style
4. Ajout de marker



7. Vue.js



1. Mis à jour dynamique du DOM
2. Exemple
 - a. `conversations = [...]`
 - b. `conversations.push(new conversation)`
 - c. `conversations.addMessage(new message)`
 - d. `conversations.updateMessage(message)`
 - e. va automatiquement mettre à jour la map, et toutes les conversations automatiquement

8. Materialize et sass

Materialize

A modern responsive front-end framework based on Material Design



```
.mymessage{  
  color: $secondary-color;  
}
```



```
.mymessage{  
  color: #F85F6A;  
}
```

Conclusion

