

Freasypoll



Fabien Franchini
Sébastien Henneberger

Plan de présentation

- Fonctionnalités implémentées
- Technologies utilisées
- Sécurité mise en place
- Démo
- Fonctionnalités restantes à implémenter / améliorations



Fonctionnalités implémentées

Un orateur peut :

- créer une poll room
- ajouter des questions à la poll room
- supprimer des questions de la poll room
- voir des statistiques «en live» de chaque question de la poll room
(nb de réponse de chaque type et nombre de réponse au total,
avec un donut chart)



Fonctionnalités implémentées

Un auditeur peut :

- accéder à une poll room via son nom et son mot de passe
- répondre une fois à une question
- choisir un choix de réponse (A, B ou C) par question
- voir le nombre d'auditeurs qui ont déjà répondu à une question



Fonctionnalités implémentées

Une question :

- a trois choix de réponses (A, B, C)
- n'a pas de réponse définie comme juste

Technologies utilisées



Angular-fullstack

- Socket.io
- Gulp
- Webpack
- Angular-chart.js

Sécurité mise en place

- Une poll room peut être protégée par un mot de passe
(pas d'exigence de complexité du mot de passe)
- Seulement les orateurs sont authentifiés
- Possibilité de s'authentifier via ses identifiants Twitter
- Les endpoints de l'API utilisés par un orateur sont sécurisés
(JWT dans le header Authorization à chaque requête)



Fonctionnalités restantes à implémenter / améliorations

Fonctionnalités restantes :

- **Un orateur devrait pouvoir fermer une poll room**
(plus possible de répondre mais les résultats seraient toujours visibles)
- **Un orateur devrait pouvoir supprimer définitivement une poll room**
- **Corriger les quelques bugs du template HTML/CSS**

Fonctionnalités restantes à implémenter / améliorations

Améliorations :

- Le contrôle qu'un auditeur ne vote qu'une fois par question est réalisé du côté client -> pas sécurisé
- Ne pas permettre la création d'une poll room si le nom utilisé est déjà existant

Des questions ?