Rapport Messagerie

La communication sous tous ses angles

Table des matieres

Presentation

Ce site web héberge un système de messagerie instantanée, c'est-à-dire il permet le dialogue entre plusieurs utilisateurs connectés à un salon.

Le principal avantage de notre “chat” est qu'il offre aux utilisateurs trois modes de communications qui sera expliqué plus tard dans le rapport.

Le but de ce site web était de mettre en place ces différents moyens de communications en parallèle tout en garantissant un fonctionnement optimal du chat.

Outils et Langages

HTML5

Utilisé pour representer les pages webs souvent combinés avec le CSS/JS.

Javascript/JQuery

Utilisé pour effectuer les différentes requêtes AJAXs et effectuer divers changement sur la page sans rafraichissement de celle-ci.

NodeJS

Framework JS orienté réseau,il nous permet de créer un serveur en Javascript.

Express

NodeJS Framework très efficace qui permet d’organiser notre application sous le modèle MVC côté serveur.
On a utilisé Express pour simplifier la mise en place des différentes routes pour le serveur.
Si on voulait mettre en place une base de données pour améliorer notre projet, Express rentre en parfaite corrélation avec des base de données comme MongoDB.

BootStrap

Utilisé uniquement pour son système de grille , Bootstrap nous aide à produire un affichage correct sur smartphone ou tablette.

Mode de communication

Polling

En francais “Attente active” est une technique de programmation que les processus utilisent lorsqu’ils vérifient de façon répétée si une condition est vraie.

Dans notre site web, le “polling” se traduit par l'envoi d’une requête au serveur toute les une seconde, le serveur retourne immédiatement une réponse s’il y a des données ou non.

Long-Polling

Fonctionnement similaire au “Polling” avec quelques variations.

Dans notre site web, le client fait une requête d’information au serveur de la même façon qu’avec le “Polling”. Par contre, si le serveur n’a pas de donnée à envoyer au client, il va se mettre en attente.Une fois l’information disponible (ou après un certain laps de temps) , le serveur va directement envoyé la réponse au client.

Push

Le Serveur Push est un mode de communication client-serveur dans lequel le dialogue est lancé par le serveur.

Dans notre site web ,dès qu’un client se connecte, une connexion est ouverte entre le serveur et lui meme, ce qui permet une mise à jour des données instantanément.

Notice d'utilisation

Nous avons choisi d'héberger notre projet sur GitHub car nous sommes familiers avec cet outil.

De plus, il s'est avéré puissant et utile pour le travail en groupe.

Les différentes étapes pour le premier lancement du "Chat" ci-dessous :

- Télécharger le zip à l'adresse : **https://github.com/hugodes/chat**
- Se placer dans le dossier extrait et faire la commande : *npm install*
- Lancer le serveur avec la commande : *node server.js*
- Aller à l'adresse : http://127.0.0.1:3000
- Communiquer avec le mode communication de votre choix

Conclusion

Ce projet nous a permis d'appréhender et comprendre les différents modes de communications possibles en utilisant AJAX et les WebSockets.

Le point important des websockets est qu'elles permettent une communication bi-directionnel, c'est-à-dire que le client ou le serveur peuvent s'envoyer mutuellement des messages. Contrairement à AJAX ou le serveur n'a aucune méthode réel "push" pour envoyer des messages au client.

Ces deux méthodes d'envoie de message permet à des applications Web tels que Gmail ou Facebook de donner à l'utilisateur la sensation d'utiliser une application bureau en temps réel.