
Équipe 302

Protocole de communication

Version 1.0

Historique des révisions

[Dans l'historique de révision, il est nécessaire d'écrire le nom du ou des auteurs ayant travaillé sur chaque version.]

Date	Version	Description	Auteur
2023-03-18	1.0	Introduction et communication client-serveur sprint 2	Alexandre Fournier
2023-03-20	1.0	Description des paquets sprint 2	Alexandre Fournier

Table des matières

[Mettre à jour la table des matières avant une remise.]

1. Introduction	4
2. Communication client-serveur	4
3. Description des paquets	4

Protocole de communication

1. Introduction

Ce document a pour objectif de présenter les protocoles de communications qui composent notre projet de développement web. Il décrit la manière dont les clients et le serveur communiquent entre eux à travers les différents services de notre application ainsi que la description de leurs protocoles de communications respectifs. Finalement, le contenu des différents paquets échangés à travers le réseau est également exposé.

2. Communication client-serveur

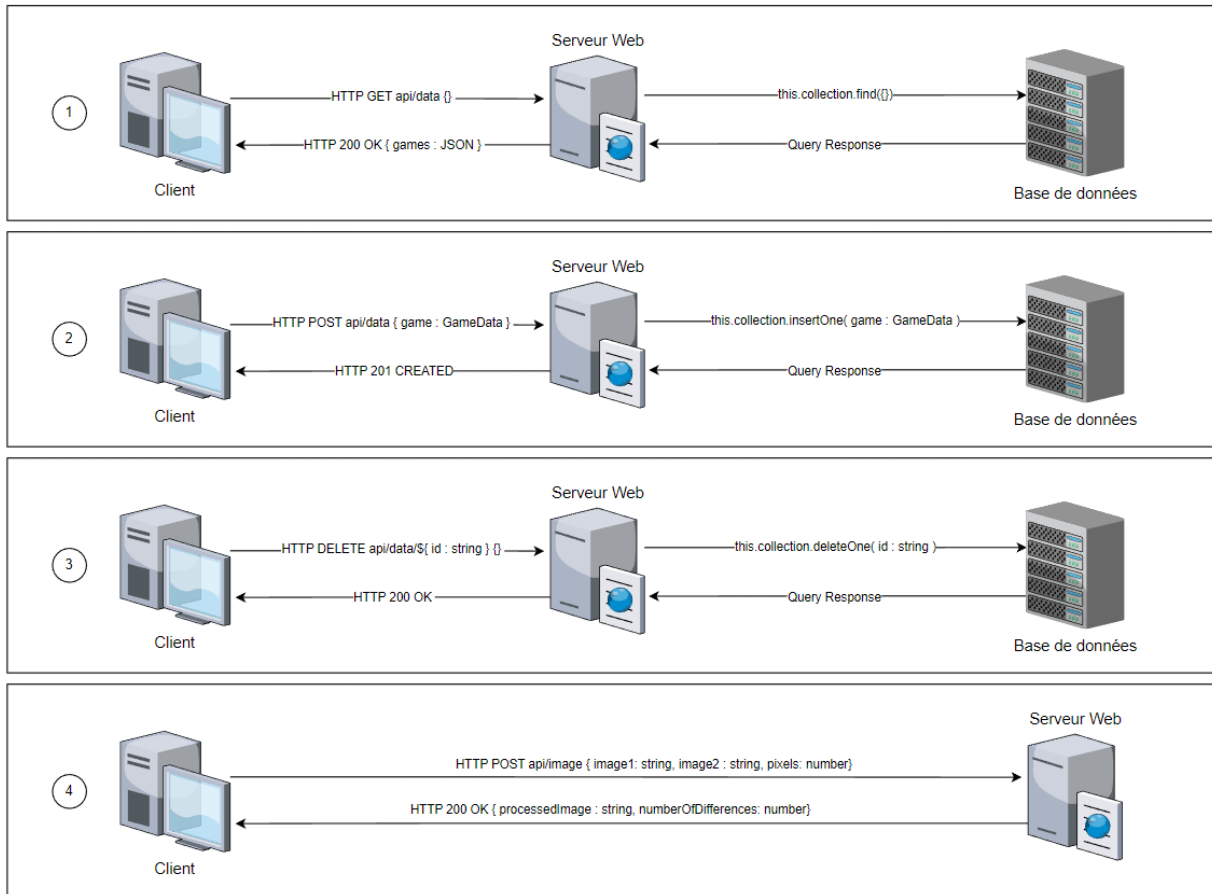
Pour ce qui est de la communication client-serveur dans notre projet, nous avons opté pour deux protocoles de communication différents afin de répondre aux besoins spécifiques de notre application et de tirer parti des avantages de chaque protocole. Les protocoles réseaux de communications qui ont été retenus sont HTTP et WebSocket.

D'abord, le HTTP a été utilisé pour tout ce qui concerne la communication avec la base de données en passant par le serveur. En effet, le client utilise le HTTP afin d'obtenir l'information relative aux différentes fiches de jeux lorsqu'il charge la page de sélection. De plus, on utilise également ce protocole pour ajouter de nouveaux jeux à la base de données ou supprimer un jeu existant. Le serveur réceptionne les requêtes du client et effectue par la suite les requêtes appropriées à la base de données MongoDB afin de la mettre à jour ou d'obtenir l'information recherchée. Le HTTP est idéal pour accéder à des données persistantes car il permet une communication asynchrone qui ne bloque pas le fil d'exécution de l'application, ce qui est particulièrement utile lors du transfert de gros volumes de données. Ensuite, le HTTP est également utilisé par le client pour le système de détection de différences. En effet, lorsqu'un client veut savoir le nombre de différences entre deux images qu'il a téléchargé sur l'application, le client envoie une requête HTTP contenant les deux images et retourne le nombre de différences ainsi qu'une représentation visuelle de celles-ci.

Ensuite, en ce qui concerne la logique du jeu, nous avons évidemment opté pour les WebSockets. Il faut savoir que chaque nouvelle partie crée une instance de la classe game sur le serveur et établit par la même occasion une connexion par Socket avec les joueurs concernés. Les WebSocket sont particulièrement adaptés à la création de parties multijoueurs et à la gestion en temps réel de l'état de la partie entre le serveur et les joueurs (clients). Grâce à SocketIO, on crée des 'rooms' qui permettent aux clients de créer des parties et de rejoindre la partie d'un autre client. De plus, l'utilisation des Sockets nous permet de facilement synchroniser en temps réel les actions des joueurs avec l'état du jeu. Le nombre de différences trouvées et le temps restant peuvent donc être synchronisés avec le serveur sans avoir à effectuer constamment des requêtes comme il aurait fallu si nous avions utilisé le HTTP. De plus, il faut savoir que ce principe de salle permet d'implémenter facilement un mécanisme de communication en temps réel entre les joueurs d'une même partie (chat). Bref, c'est grâce à la rapidité de transmission des données des WebSockets et à leur capacité de synchronisation client-serveur en temps réel que nous pouvons fournir une expérience de jeu fluide et réactive.

3. Description des paquets

Description des Paquets - Protocole HTTP

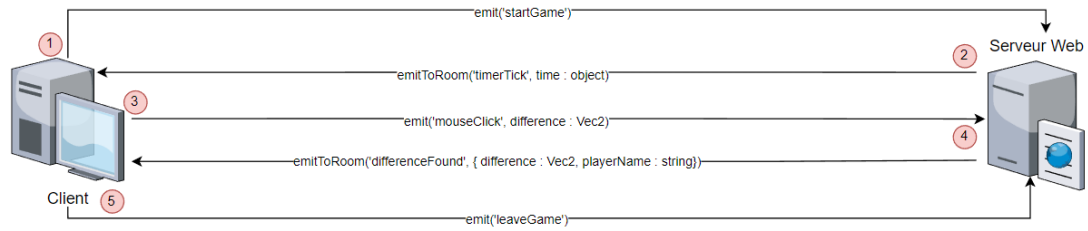


Description des Paquets - Protocole WebSockets



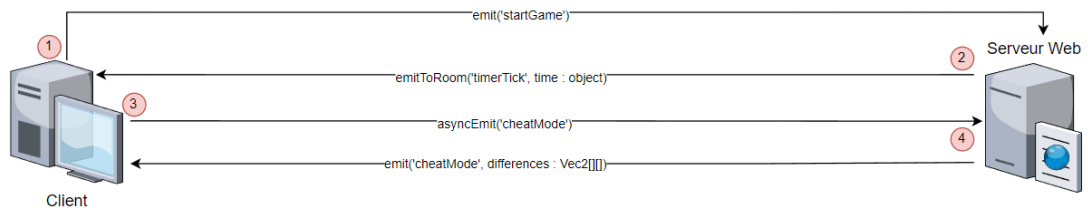
3

Mode de jeu solo (abandon)



4

Mode triche



5 Mode de jeu 1 contre 1



6 Utilisation du chat (Multijoueur)

