Équipe 11

Projet Intégrateur de 3e Année Poly Paint
Protocole de communication

Version 2.0

Historique des révisions

Date	Version	Description	Auteur
2019-01-29	1.1	Proposition de protocole	Nicolas Coutu
2019-02-06	1.2	Description des paquets	Nicolas Coutu, Tom Gautier
2019-02-07	1.3	Communication client-serveur	Tom Gautier
2019-02-09	2.0	Revue finale	Tom Gautier

Table des matières

1. Introduction **4**

2. Communication client-serveur **4**

3. Description des paquets **5**

3.1. Connection au serveur 5
3.1. Déconnexion du serveur 5
3.2. Authentification 5
3.3. Chat - Envoyer un message 5
3.4. Chat - Joindre un canal de discussion 5
3.5. Chat - Créer un canal de discussion 6
3.6. Chat - Inviter un usager à un canal de discussion 6
3.7. Galerie - Joindre une session collaborative 6
3.8. Galerie - Créer une session collaborative 6
3.9. Galerie - Aimer une image 6
3.10. Galerie - Ne plus aimer une image 7
3.11. Galerie - Obtenir les images des galeries 7
3.12. Accessibilité des images - Édition de l'accès 7
3.11. Liste d'utilisateur - Obtenir la liste des utilisateurs en ligne 7
3.12. Liste d'utilisateur - Obtenir tous les utilisateurs 7
3.13. Amis - Ajouter un ami 8
3.14. Amis - Supprimer un ami 8
3.14. Accueil - Obtenir les nouvelles images publiées 8
3.15. Tags - Ajouter des tags 8
3.16. Tags - Supprimer des tags 8
3.17. Session collaborative - Ajouter un élément 9
3.18. Session collaborative - Retirer un ou des éléments 9
3.19. Session collaborative - Modifier un élément 9
3.20. Session collaborative - Sélectionner un ou des éléments 9
3.21. Session collaborative - Annuler la sélection 9
3.22. Session collaborative - Redimensionner le canevas 10

Protocole de communication

1. Introduction

Ce document présente les décisions prises en ce qui concerne la partie réseau de la solution Poly Paint. Tout d'abord, une brève explication des différents moyens utilisés ainsi que de leur utilisation dans le programme sera faite.
Par la suite, la description des paquets nécessaires à la communication dans l'application sera détaillée à travers plusieurs tableaux, associés en paire Requête-Réponse.

2. Communication client-serveur

La majorité de la communication à travers l'application se fait en temps réel et avec une grande fluidité. En effet, que ce soit dans le clavardage ou lors d'une session collaborative, on veut établir une communication avec le moins de latence possible, permettant ainsi une expérience utilisateur agréable.
Par conséquent, nous avons choisi d'utiliser les websockets pour ces applications. Cette technologie possède des avantages qui nous intéressent particulièrement dans le cadre du projet PolyPaint. Parmi les principales caractéristiques, elle permet:

- Un temps de réaction très rapide: étant donné que les WebSockets n'ont pas besoin du "overhead" classique des requêtes/réponses HTTP pour chaque message reçu et envoyé, la communication devient plus efficiente;
- De gérer le "long-polling" d'HTTP d'une manière élégante afin que les clients n'aient pas à faire de requêtes régulières au serveur pour rafraîchir les informations dont ils ont besoin. Celles-ci leur seront directement communiquées lors d'un changement ou d'une modification;
- Le support pour les "salles" est intégré. Cela facilitera grandement le développement du serveur afin de gérer de multiples canaux de discussion et sessions collaboratives. Cela se fait de manière très efficace et avec une latence faible.

Néanmoins, l'usage du protocole HTTP sera justifiée dans plusieurs fonctionnalités de l'application, notamment celles dont les mises-à-jour ne seront pas aussi fréquentes. En effet, ce protocole reconnu pour sa stabilité est très utile pour :

- Demander de l'information à la base de données que l'on n'aura pas besoin de mettre à jour le plus rapidement possible;
- Gérer des requêtes qui auront possiblement des erreurs, comme l'authentification à un logiciel par exemple.

3. Description des paquets

Tous les tableaux de cette section respecteront la nomenclature:

Raison de la requête	Protocole	Événement	Arguments

Tous les arguments des requêtes seront communiqués en format JSON.

3.1. Connection au serveur

Requête au serveur	Websockets	Connection	-

3.1. Déconnexion du serveur

Requête au serveur	Websockets	Disconnect	-

3.2. Authentification

Requête au serveur	HTTP	POST	Username: string, password: string
Réponse du serveur	HTTP	SEND	200: Succès 400: Mauvaise requête 403: Accès refusé 520: Unknown error

3.3. Chat - Envoyer un message

Requête au serveur	Websockets	MessageSent	Date: string, author: string, message: string, roomId: string
Réponse du serveur	Websockets	MessageReceived	Date: string, author: string, message: string, roomId: string

3.4. Chat - Joindre un canal de discussion

Requête au serveur	Websockets	JoinChannel	channelName: string, username: string
Réponse du serveur	Websockets	ChannelJoined	chatHistory: Message[ ], channelId: string
Réponse du serveur	Websockets	ChannelAlreadyJoined	

3.5. Chat - Créer un canal de discussion

Requête au serveur	Websockets	CreateChannel	channelName: string
Réponse du serveur	Websockets	ChannelCreated	channelId: string

3.6. Chat - Inviter un usager à un canal de discussion

Requête au serveur	Websockets	InviteUserToChannel	channelName: string, username: string
Réponse du serveur	Websockets	UserInvitedToChannel	-
Réponse du serveur	Websockets	UserAlreadyInChannel	-

3.7. Galerie - Joindre une session collaborative

Requête au serveur	Websockets	JoinSession	imageID : string, password (opt.) : string
Réponse du serveur	Websockets	SessionJoined	image: Image

3.8. Galerie - Créer une session collaborative

Requête au serveur	Websockets	CreateSession	name: string, protection: Protection, visibility: Visibility, image: Image
Réponse du serveur	Websockets	SessionCreated	success : bool

3.9. Galerie - Aimer une image

Requête au serveur	HTTP	PUT	imageID : string
Réponse du serveur	HTTP	SENT	newLikesNumber: int 400: Mauvaise requête 403: Accès refusé 520: Unknown error

3.10. Galerie - Ne plus aimer une image

Requête au serveur	HTTP	PUT	imageID : string
Réponse du serveur	HTTP	SEND	newLikesNumber: int 400: Mauvaise requête 403: Accès refusé 520: Unknown error

3.11. Galerie - Obtenir les images des galeries

Requête au serveur	HTTP	GET	images: string = ‘images’
Réponse du serveur	HTTP	SEND	images: Image[] 400: Mauvaise requête 403: Accès refusé 520: Unknown error

3.12. Accessibilité des images - Édition de l’accès

Requête au serveur	Websockets	UpdateAccess	protection: Protection, visibility: Visibility
Réponse du serveur	Websockets	AccessUpdated	success : bool

3.11. Liste d’utilisateur - Obtenir la liste des utilisateurs en ligne

Requête au serveur	Websockets	GetOnlineUsers	-
Réponse du serveur	Websockets	OnlineUsersList	UserList : Users[]

3.12. Liste d’utilisateur - Obtenir tous les utilisateurs

Requête au serveur	HTTP	GET	users: string = ‘users’
Réponse du serveur	HTTP	SEND	Users: User[] 400: Mauvaise requête 403: Accès refusé 520: Unknown error

3.13. Amis - Ajouter un ami

Requête au serveur	Websockets	AddFriend	username : string
Réponse du serveur	Websockets	FriendAdded	success : bool

3.14. Amis - Supprimer un ami

Requête au serveur	Websockets	RemoveFriend	username : string
Réponse du serveur	Websockets	FriendRemoved	success : bool

3.14. Accueil - Obtenir les nouvelles images publiées

Requête au serveur	HTTP	GET	IMAGE_NUMBER: string
Réponse du serveur	HTTP	SEND	images: Image [IMAGE_NUMBER] 400: Mauvaise requête 403: Accès refusé 520: Unknown error

3.15. Tags - Ajouter des tags

Requête au serveur	HTTP	POST	tags : string[], imageId: string
Réponse du serveur	HTTP	SEND	200: Succès 400: Mauvaise requête 403: Accès refusé 520: Unknown error

3.16. Tags - Supprimer des tags

Requête au serveur	HTTP	REMOVE	tags : string[], imageId: string
Réponse du serveur	HTTP	SEND	200: Succès 400: Mauvaise requête 403: Accès refusé 520: Unknown error

3.17. Session collaborative - Ajouter un élément

Requête au serveur	Websockets	AddElement	ElementProperties: Properties, elementType: ElementType
Réponse du serveur	Websockets	ElementAdded	roomId : string, ElementProperties: Properties, elementType: ElementType

3.18. Session collaborative - Retirer un ou des éléments

Requête au serveur	Websockets	DeleteElements	ElementIds : string[]
Réponse du serveur	Websockets	ElementsDeleted	roomId : string, elementIds: string[]

3.19. Session collaborative - Modifier un élément

Requête au serveur	Websockets	ChangeElement	ElementId: string, newElementProperties: Properties, ElementType: ElementType
Réponse du serveur	Websockets	ElementChanged	roomId : string

3.20. Session collaborative - Sélectionner un ou des éléments

Requête au serveur	Websockets	SelectObjects	ElementId : string[]
Réponse du serveur	Websockets	ObjectsSelected	roomId : string, elementId: string

3.21. Session collaborative - Annuler la sélection

Requête au serveur	Websockets	UnselectObjects	ElementId : string[]
Réponse du serveur	Websockets	ObjectsUnselected	roomId : string, elementId: string

3.22. Session collaborative - Redimensionner le canevas

Requête au serveur	Websockets	ChangeCanvas	width : int, height : int
Réponse du serveur	Websockets	CanvasChanged	roomId : string, width : int, height : int