

Protocole de communication

Client - Serveur

Introduction

Ce document détail le protocole de communication client-serveur de l'application.

Connexion

Client

La connexion à un débat s'effectue grâce à l'utilisation d'un namespace spécifique lors de la connexion. Le namespace est /DEBATE-%id du débat%.

L'argument path vaut /socket-io, et le paramètre du handshake est le suivant :

- uuid: Id du client

Admin

La connexion s'effectue grâce aux paramètres du handshake sur le namespace /**admin** avec les paramètres suivants:

- username: Nom d'utilisateur
- password: Mot de passe

Questions

Message	De	Description	Structure	Réponse
getQuestions	Client	Demande les questions du débat	-	Array de questions du débat
answerQuestion	Client	Réponse à une question fermée	questionId: Id de la question answerId: Id de la réponse	Boolean si l'opération à fonctionnée
answerOpenQuestion	Client	Réponse à une question ouverte	questionId: Id de la question answer: Réponse sous forme de string	Boolean si l'opération à fonctionnée
getSuggestedQuestions	Client	Liste des questions suggérées	Id:id du débat	Array de { suggestionId: id de la suggestion, question : question posée, votes: nombre de votes }
suggestQuestion	Client	Suggestion d'une nouvelle question	question: Question suggérée	False en cas d'échec, id de la suggestion sinon
voteSuggestedQuestion	Client	Vote pour une question suggérée	suggestionId: Id de la suggestion	Boolean si l'opération à fonctionnée

suggestedQuestion	Server	Notifie les clients qu'une nouvelle suggestion est disponible	{ suggestionId: id de la suggestion, question : question posée, votes: nombre de votes }	-
newQuestion	Server	Notifie les clients qu'une nouvelle question est disponible	question: la question	-

Fonction d'un administrateur

Message	De	Description	Structure	Réponse
getDebates	Client	Demande la liste des débats existants.	-	Array de { debateld : id du débat title : titre du débat description : description du débat Closed : indique si le débat est fermé ou ouvert }
getDebateDetails	Client	Demande les détails d'un débats.	debateld: Id du débat	Array de { debateld : id du débat title : titre du débat description : description du débat startTime: début du débat finishTime: Fin du débat Closed : indique si le débat est fermé ou ouvert }
getDebateSuggestions	Client	Demande les suggestions d'un	debateld: Id du débat	Array de { Array de {

		débat		Uuid: uuid du clients ayant suggéré suggestionId: id de la suggestion suggestion: texte de la suggestion votes: nombre de votes pour la suggestion Removed: permet de savoir si la suggestion a été supprimée(refusée) } }
getDebateQuestions	Client	Demande la liste des questions liées à un débat.	debateId: Id du débat	
newDebate	Client	Demande la création d'un nouveau débat	Objet débat contenant { title: titre du débat description: description du débat }	ID du débat -1 si l'opération n'a pas fonctionné
closeDebate	Client	Termine un débat	aldDiscussion: Id du débat	True si la fermeture c'est déroulée correctement false sinon
lockDebate	Client	Permet de bloquer la connexion à un débat	debateId: id du débat	True si le débat à pu être fermé faux sinon
unlockDebate	Client	Permet de débloquent un débat	debateId: id du débat	True si le débat à pu être débloquent faux sinon
newQuestion	Client	Création d'une nouvelle question et	debateId : id du débat	ID de la question

		transmission aux clients.	Objet Question contenant { debateId: id du débat lié à la question title: titre de la question answers: tableau des réponses possibles isOpenQuesti on: booléan insiquant si la question est ouverte }	-1 si l'opération n'a pas fonctionnée sinon l'id de la question est renvoyé
banUser	Client	Bannit un auditeur des débats d'un administrateur	Objet contenant { uuid: Identifiant de l'utilisateur debateId :id du débat, }	True si le ban a été effectué False si cela n'est pas le cas
unbanUser	Client	Annule le bannissement d'un auditeur.	uuid: Identifiant de l'utilisateur	True si le ban à pu être levé false sinon
approveQuestion	Client	Approuve la suggestion d'un auditeur.	Objet contenant { debateId: Id du débat questionId: Id de la question }	True si le question a été approuvée false sinon
rejectQuestion	Client	Rejete la suggestion	Objet	

		d'un auditeur	contenant { debatId: id du débat suggestionId: id de la suggestion	
--	--	---------------	---	--

Message serveur-client administrateur

Message	De	Description	Donnée envoyée
banned	Serveur	Permet d'informer le clients qu'il a été banni	Message indiquant que l'utilisateur a été banni

Statistique administrateur

Message	De	Description	Structure	Réponse
getAdminStats	Client	Demande les statistique de débat pour un admin	Username: username de l'admin	Array de { numberDebates: nombre de débat, Array de débat{ debatId : id du débat title : titre du débat description : description du débat, Time: temps total du débat. } }
getDebatStats	Client	Demande les statistique	debatId: id	Array de {

		d'un débats. Les questions seront ordonnées dans l'ordre décroissant(question avec plus de vote en première)	du débat	<p>numberQuestions: nombre de questions pour le débat</p> <p>numberParticipants : nombre de participant</p> <p>Array de questions{ title: titre de la question</p> <p>numberVotes: nombre de vote total pour la question } }</p>
getQuestionStats	Admin	Demande les statistiques d'une question.	Objet contenant { questionId: id de la question discussionId: Id de la discussion lié à la question }	<p>Array de {</p> <p> numberResponses: nombre de réponse pour la question,</p> <p> interest: Pourcentage de réponse par rapport au nombre de participant.</p> <p> Array de {</p> <p> response: Réponse donnée pour la question</p> <p> Percentage: Pourcentage obtenu pour la réponse.</p> <p> }</p> <p>}</p>

Suggestion utilisateur

Message	De	Description	Donnée envoyée
newVote	Serveur	Permet d'informer le clients et l'administrateur qu'une suggestion a été votée	Id de la suggestion votée
suggestedQuestion	Serveur	Permet d'indiquer au client et au administrateur qu'une nouvelle suggestion est disponible	Objet suggestion contenat:{ suggestionId: id de la suggestion suggestion: suggestion proposée votes: nombre de votes pour la suggestion Remove: si la suggestion a été enlevé(invalidée) par l'administrateur }
rejectedSuggestion	Serveur	Indique à un administrateur qu'une suggestion a été refusée	suggestionId: id de la suggestion
deletedSuggestion	Serveur	Indique au client et à l'administrateur q'une suggestion a été supprimée	suggestionId: id de la suggestion

Fonctions clientes

Message	De	Description	Structure	Réponse
getDebateDetails	Client	Demande les informations du débat en cours	-	Array de { debateId : id du débat title : titre du débat description : description du débat }
getQuestions	Client	Retourne la liste des questions du débat en cours. Les questions sont marquées afin de savoir si l'utilisateur a répondu	-	Array de { id : id de la question, title: titre de la question, answers: tableaux de réponses, isOpenQuestion : booléen indiquant si la question est ouverte, answered: booléen indiquant si la question est répondue }
answerQuestion	Client	Répond à une question fermée du débat	{ questionId: id	Booléen si l'opération a

			de la question, answerId: id de la réponse }	réussi
answerOpenQuestion	Client	Répond à une question ouverte du débat	{ questionId: id de la question, answer: réponse à la question }	Booléen si l'opération a réussi
getSuggestedQuestions	Client	Retourne la liste des suggestions active du débat et marque les suggestions non-votées	-	Array de { suggestionId: id de la suggestion, suggestion: question suggérée, votes: le nombre de votes, voted: booléen indiquant si elle a déjà été votée }
suggestQuestion	Client	Suggère une nouvelle question	suggestion: question suggérée	id de la suggestion si l'opération a fonctionné, false sinon
voteSuggestedQuestion	Client	Vote pour une suggestion du débat	suggestionId : id de la suggestion	True si l'opération a fonctionné, false sinon