



Polygram
Protocole de communication

Version 1.0

Historique des révisions

Date	Version	Description	Auteur
2022-02-11	1.0	Rédaction initiale du protocole de communication	Équipe 203

Table des matières

1. Introduction	4
2. Communication client-serveur	4
3. Description des paquets	5
3.1 Description des paquets pour la communication via les sockets	5
3.1.1 Clavardage	5
3.1.1.1 Transmission de message dans les canaux de discussion	5
3.1.1.2 Créer, supprimer et rejoindre d'un nouveau canal de discussion	5
3.1.2 Connexion/déconnexion	5
3.1.3 Notifications	5
3.1.4 laisser une Mention « j'aime » sur un dessin	6
3.2 Description des paquets pour la communication via REST API	6
3.2.1 authentification	6
3.2.1.1 connexion à un compte existant (log In)	6
3.2.1.2 création d'un nouveau compte (Sign In)	7
3.2.1.3 Déconnexion d'un compte existant (Log out)	8
3.2.2 Statistiques du compte	9
3.2.3 Changement d'avatar	10
3.2.4 Notifications	11
3.2.4.1 Obtenir toutes les notifications associées à un utilisateur	11
3.2.4.2 Supprimer une notification après la réponse	11
3.2.5 Dessin	11
3.2.5.1 Créer un dessin	11
3.2.5.2 Supprimer un dessin	12
3.2.6 Gestion des albums	13
3.2.6.1 Création d'un album	13
3.2.6.2 Suppression d'un album	14
3.2.6.3 Changement des attributs d'un album	14
3.2.6.4 Rejoindre un album existant	16
3.2.6.3 Liste des albums	16
3.2.7 Accès à un profil	17
3.2.8 Story	17
3.2.8.1 Visualiser une story	18
3.2.8.2 Supprimer une story	18
3.2.9 Téléversement d'un dessin sur Google Drive	19
3.2.9 Téléversement d'un dessin sur Dropbox	19
3.2.9 Modifier un effet visuel	20

Protocole de communication

1. Introduction

Le document suivant a pour objectif de détailler les interactions entre les clients et le serveur. Il vise à présenter les flux de communications et les différentes qui seront appliquées à l'application Polygram. D'une part, il détaille la communication client-serveur. D'une autre part, il détaille les différents types de paquets qui y seront utilisés.

2. Communication client-serveur

Deux protocoles seront utilisés dans Polygram : WebSocket et HTTP.

D'une part, la librairie Socket.io sera utilisée tant sur le serveur que les clients (lourd et léger). Cette librairie utilise le protocole WebSocket. Ce protocole permet une communication bidirectionnelle entre le client et le serveur. De plus, elle assure une réduction du temps de latence entre les différents utilisateurs connectés. Les fonctionnalités utilisant ce protocole sont :

- Le clavardage
- Le dessin collaboratif
- Les notifications à la réception d'un nouveau message

D'une autre part, l'utilisation du protocole HTTP sera sollicitée. En effet, plusieurs autres fonctionnalités de Polygram ne demandent pas un envoi continu de données et peuvent se faire de manière unidirectionnelle (client vers serveur). Les requêtes HTTP seront envoyées en utilisant le REST API. Ces fonctionnalités sont:

- L'authentification des utilisateurs
- La création de comptes
- Le changement de mot de passe
- La gestion d'avatar
- La collecte de statistiques et le système de badge
- L'accès à d'autres profils
- L'accès à son propre profil
- L'accès aux dessins d'un album
- La liste d'album
- Les *stories* de dessin
- Les mentions "j'aime"
- La création et la suppression de dessin
- La création d'un album
- Le changement des attributs d'un album
- Le changement des attributs d'un dessin
- L'envoi d'un dessin sur un système d'hébergement infonuagique.
- La sauvegarde des effets visuels sonores

3. Description des paquets

Dans cette section, nous allons détailler en premier lieu les événements impliqués dans la communication via Sockets, ensuite les requêtes REST API, le tout, en décrivant le type de données envoyées et reçues.

Les fonctionnalités seront réparties entre les deux moyens de communication comme nous l'avons présenté dans la section précédente.

3.1 Description des paquets pour la communication via les sockets

3.1.1 Clavardage

3.1.1.1 Transmission de message dans les canaux de discussion

Évènement: chat	Données: { username, channelID, message }
-----------------	---

Cet évènement consiste en l'envoi et la réception de messages dans un canal de discussion. Lors de cette opération, le paquet transmis sera **un message** ayant les propriétés suivantes :

Attribut	Type	Description
username	String	Auteur du message
channelID	String	Canal d'où provient le message
message	String	Message envoyé
timestamp	Timestamp	Instant où le message est reçu par le client

3.1.1.2 Créer, supprimer et rejoindre un nouveau canal de discussion

Évènement: createChannel, deleteChannel, joinChannel	Données: { channelID: string, username:string, confirmation: bool, Reason: string }
--	---

Du client vers le serveur :

Attribut	Type	Description
username	String	Auteur du message qui initie le canal
channelID	String	Canal de discussion concerné

Du serveur vers le client :

Attribut	Type	Description
confirmation	bool	Confirmation de création, suppression, jointure
reason	String	Raison de l'échec en cas d'échec

3.1.2 Connexion/déconnexion

Évènement: login	Données: { username, password (encodé) }
------------------	--

Évènement: logout	Données: { username, password (encodé) }
-------------------	--

3.1.3 Notifications

Attribut	Type	Description
Receiver	String	Identifiant de l'utilisateur qui reçoit la notification

Message	String	Message qui sera reçu
---------	--------	-----------------------

3.1.4 Dessin collaboratif

Attribut	Type	Description
username	String	L'utilisateur qui dessine
stroke	Stroke	Un trait dessiné
creatorID	String	Le propriétaire du dessin
drawingID	String	Identifiant du dessin

Avec Stroke qui aurait comme attributs :

Attribut	Type	Description
drawingID	String	Identifiant du dessin concerné
order	Int	Ordre de la séquence des traits
Start_pos	Int[]	Position initiale en x et y
End_pos	Int[]	Position finale en x et y
color	rgb(int,int,int)	Couleur du trait

3.1.4 Laisser une mention « j'aime » sur un dessin

Attribut	Type	Description
drawingID	String	Identifiant du dessin
likeFlag	Bool	Une mention j'aime met le flag à true

3.2 Description des paquets pour la communication via REST API

3.2.1 Authentification

3.2.1.1 connexion à un compte existant (login)

Attribut	Type	Description
URL	~/users/login	
Requête	POST	
Request body		
username	String	Nom unique de l'utilisateur

password	String	Mot de passe encodé
Valid response		
Status	200	Authentification acceptée
Avatar	String	Image en Base64
Invalid response		
Status	401	Connexion non autorisée

3.2.1.2 création d'un nouveau compte (Sign In)

Attribut	Type	Description
URL	~/users/create	
Requête	POST	
Request body		
profileInfo	IprofileInfo	interface contenant les informations relatives à l'utilisateur
Valid response		
Status	201	Utilisateur créé
Invalid response		
Status	409	Conflit: Le nom d'utilisateur est déjà utilisé

Avec l'interface IProfileInfo contenant les informations suivantes :

Attribut	Type	Description
username	String	Nom unique de l'utilisateur
Email adress	String	Adresse email (pour récupération de mot de passe éventuellement)
Password	String	Mot de passe encodé
FirstName	String	Prénom du joueur
LastName	String	Nom du joueur

Avatar	String	Image en Base64 (soit prédéfinies ou prise par la caméra)
--------	--------	---

3.2.1.3 Déconnexion d'un compte existant (Log out)

Attribut	Type	Description
URL	~/users/:username/logout	
Requête	PATCH	
Request header		
Authorization	String	Bearer token de l'utilisateur
username	String	Nom de l'utilisateur qui veut se déconnecter
Valid response		
Status	200	L'utilisateur a été déconnecté.
Request header		
Status	401	Le token a été refusé.

3.2.1.4 Changement d'un mot de passe oublié

Attribut	Type	Description
URL	~/users/:username/password	
Requête	PUT	
Request header		
Authorization	String	Bearer token de l'utilisateur
username	String	Nom de l'utilisateur qui veut se déconnecter
password	String	Nouveau mot de passe
Valid response		
Status	200	OK : le mot de passe a été modifié

Request header		
Status	401	Le token a été refusé.

3.2.2 Statistiques du compte

Attribut	Type	Description
URL	~/users/:username/general-statistics	
Requête	GET	
Request header		
Authorization	String	Bearer token pour donner accès à la page
username	String	Nom d'utilisateur dont les statistiques nous intéressent
Valid response		
Status	200	OK
username	string	Identifiant de l'utilisateur
generalStats	IGeneralStats	Les statistiques générales concernant le profil de l'utilisateur
advancedStats	IAdvancedStats	Les statistiques spécifiques concernant le profil de l'utilisateur
Invalid response		
Status	400	Aucun utilisateur correspondant n'a été trouvé.

Ou IGeneralStats contient les attributs suivants :

Attribut	Type	Description
nbCollabDrawings	Int	Nombre de dessins auxquels il a collaboré
nbOwnedDrawings	Int	Nombre de dessins auxquels il est auteur
nbPrivateAlbums	Int	Nombre d'albums privés dont il est membre

avgCollabTime	Int	Temps moyen de ses séances de collaboration
totalCollabTime	Int	Temps total passé à collaborer

Et IAdvancedStats :

Attribut	Type	Description
Badge	Enum: Beginner Intermediate Expert Artist	Type de badge
totalReceivedlikes	Int	Nombre total de mentions j'aime reçues
bestDrawing	List<Drawing>	Top 3 Dessins les plus aimés

3.2.3 Changement d'avatar

Attribut	Type	Description
URL	~/users/:username/avatar	
Requête	PATCH	
Request header		
Authorization	String	Bearer token pour donner accès à la page
username	String	Nom d'utilisateur concerné par le changement
Request body		
Avatar	String	Nouvel avatar en Base64
Valid response		
Status	200	OK
Invalid response		
Status	401	Accès refusé à la ressource.

3.2.4 Notifications

3.2.4.1 Obtenir toutes les notifications associées à un utilisateur

Attribut	Type	Description
URL	~/users/:username/notifications	
Requête	GET	
Request body		
Valid response		
Status	200	OK
Body	Collection de notifications	Toutes les notifications de l'utilisateur.
Invalid response		
Status	404	Aucune notification trouvée

3.2.4.2 Supprimer une notification après la réponse

Attribut	Type	Description
URL	~/users/:username/notifications/:id	
Requête	DELETE	
Request body		
Valid response		
Status	200	Notification supprimée
Invalid response		
Status	404	Notification non trouvée sur la base de données

3.2.5 Dessin

3.2.5.1 Créer un dessin

Attribut	Type	Description
URL	~/drawings/:id	
Requête	POST	
Request body		
drawing	Drawing	Le dessin à enregistrer
Valid response		
Status	201	Dessin créé avec succès
Invalid response		
Status	409	Erreur

3.2.5.2 Supprimer un dessin

Attribut	Type	Description
URL	~/drawings/:id	
Requête	DELETE	
Request body		
drawing	Drawing	Le dessin à supprimer
Valid response		
Status	200	Dessin supprimé
Invalid response		
Status	404	Le dessin est introuvable

3.2.5.3 Modifier les propriétés d'un dessin

Attribut	Type	Description
----------	------	-------------

URL	~/drawings/:id	
Requête	PUT	
Request body		
drawing	Drawing	Le dessin à modifier
Valid response		
Status	200	Dessin supprimé
Invalid response		
Status	404	Le dessin est introuvable

Ou Drawing possède les attributs suivants :

Attribut	Type	Description
IdDrawing	String	Identifiant du dessin
Strokes	Vecteur de Stroke (voir le type Stroke dans la section dessin collaboratif)	Vecteur de tous les traits constituant l'image
creationTime	Timestamp	Moment de la création du dessin
idCreator	String	Identifiant du propriétaire du dessin
nbcollab	int	Nombre de collaborateurs actifs

3.2.6 Gestion des albums

3.2.6.1 Création d'un album

Attribut	Type	Description
URL	~/albums/:id	
Requête	POST	

Request body		
album	Album	Nouvel album créé
Valid response		
Status	201	Album crée avec succès
Invalid response		
Status	409	Erreur

3.2.6.2 Suppression d'un album

Attribut	Type	Description
URL	~/albums/:id	
Requête	DELETE	
Request body		
album	Album	Album à supprimer
Valid response		
Status	200	album supprimé
Invalid response		
Status	404	L'album est introuvable

3.2.6.3 Changement des attributs d'un album

Changement du nom de l'album :

Attribut	Type	Description
URL	~/albums/:album/name	
Requête	PATCH	
Request header		

Authorization	String	Bearer token pour donner accès à la page (protection dans le cas où le modificateur n'est pas le propriétaire de l'album)
albumId	String	Identifiant de l'album concerné par le changement
Request body		
name	String	Nouveau nom d'album
Valid response		
Status	200	OK
Invalid response		
Status	401	Accès refusé à la ressource.

Changement de la description de l'album :

Attribut	Type	Description
URL	~/albums/:album/description	
Requête	PATCH	
Request header		
Authorization	String	Bearer token pour donner accès à la page (protection dans le cas où le modificateur n'est pas le propriétaire de l'album)
Album	String	Identifiant de l'album concerné par le changement
Request body		
description	String	Nouvelle description de l'album
Valid response		
Status	200	OK

Invalid response		
Status	401	Accès refusé à la ressource.

3.2.6.4 Rejoindre un album existant

3.2.6.3 Liste des albums

Attribut	Type	Description
URL	~/albums	
Requête	GET	
Request body		
Valid response		
Status	200	OK
Body	List<Album>	Retourner tous les albums
Invalid response		
Status	404	Aucune notification trouvée

Avec Album contenant les attributs suivant:

Attribut	Type	Description
----------	------	-------------

albumId	String	Identifiant du dessin
isPublic	Bool	Accessibilité de l'album deux valeurs possibles: public ou private
name	String	Nom de l'album
description	String	Description de l'album
creatorId	String	Identifiant du propriétaire du dessin
drawingList	Liste de Drawing	Liste des dessins contenus dans l'album

3.2.7 Accès à un profil

Attribut	Type	Description
URL	~/users/:username/profile	
Requête	GET	
Request header		
userInfo	IProfileInfos	Informations du profil de l'utilisateur
Valid response		
Status	200	OK
Request header		
Status	404	Les informations de profil n'existent pas ou ne sont plus disponibles.

3.2.8 Story

3.2.8.1 Visualiser une story

Attribut	Type	Description
URL	~/users/:storyid	
Requête	GET	
Request header		
storyid	int	Identifiant unique de la story
story	String	Image en Base64
Valid response		
Status	200	OK
Request header		
Status	404	La story n'existe pas ou n'est plus disponible.

3.2.8.2 Supprimer une story

Attribut	Type	Description
URL	~/users/:storyid	
Requête	DELETE	
Request header		
storyid	int	Identifiant unique de la story
story	String	Image en Base64
Valid response		
Status	200	OK: la story a été supprimée avec succès
Request header		

Status	404	La story n'existe pas ou n'est plus disponible.
--------	-----	---

3.2.9 Téléversement d'un dessin sur Google Drive

Attribut	Type	Description
URL	~/upload/google-drive	
Requête	POST	
Request header		
driveusername	String	Identifiant Google Drive de l'utilisateur
drivePassword	String	Mot de passe Google Drive de l'utilisateur
drawing	Drawing	Dessin à téléverser
Valid response		
Status	200	OK : téléversement réussi
Request header		
Status	401	Accès refusé : mauvais credentials

3.2.9 Téléversement d'un dessin sur Dropbox

Attribut	Type	Description
URL	~/upload/dropbox	
Requête	POST	
Request header		
dropboxusername	String	Identifiant Dropbbox de l'utilisateur
dropboxPassword	String	Mot de passe Dropbbox de l'utilisateur

drawing	Drawing	Dessin à téléverser
Valid response		
Status	200	OK : téléversement réussi
Request header		
Status	401	Accès refusé : mauvais credentials

3.2.9 Modifier un effet sonore

Attribut	Type	Description
URL	~/users/:username/sound	
Requête	GET	
Request header		
soundid	int	Identifiant unique du son
Valid response		
Status	200	OK
Request header		
Status	404	Le son n'existe pas ou n'est plus disponible.

3.2.9 Modifier un effet visuel

Attribut	Type	Description
URL	~/users/:username/animation	
Requête	GET	
Request header		
animationid	int	Identifiant unique de l'animation

Valid response		
Status	200	OK
Request header		
Status	404	L'animation n'existe pas ou n'est plus disponible.