

Appendix B:

The following text provides an overview of the GraphQL API, detailing its schema: types, inputs, and enums, and its operations: implemented queries and mutations.

Query

Queries

FIELDS

None

- [getCurrentUserInfo](#) [User!](#)
User and Authentication Queries
- [getUser](#) ([userId](#) [ID!](#)) [User](#)
- [listUsers](#) ([query](#) [String](#) , [page](#) [Int](#) , [size](#) [Int](#)) [UserPage!](#)
- [getDialogs](#) ([page](#) [Int](#) , [size](#) [Int](#)) [DialogInfoPage!](#)
Messages Queries
- [getMessagesByUserId](#) ([userId](#) [ID!](#) , [page](#) [Int](#) , [size](#) [Int](#)) [MessagePage!](#)
- [listGroups](#) ([query](#) [String](#) , [page](#) [Int](#) , [size](#) [Int](#)) [UserPage!](#)
Groups Queries
- [getGroup](#) ([groupId](#) [ID!](#)) [Participant!](#)
- [getGroupProfile](#) ([groupId](#) [ID!](#)) [GroupProfile!](#)
- [listGroupMembers](#) ([groupId](#) [ID!](#) , [page](#) [Int](#) , [size](#) [Int](#)) [UserPage!](#)
- [listGroupPosts](#) ([groupId](#) [ID!](#) , [page](#) [Int](#) , [size](#) [Int](#)) [PostPage!](#)
- [listGroupJoinRequests](#) ([groupId](#) [ID!](#) , [page](#) [Int](#) , [size](#) [Int](#) , [sort](#) [\[String\]](#)) [RequestPage!](#)
- [getImage](#) ([imageId](#) [ID!](#)) [Image!](#)
Images Queries
- [listUserImages](#) ([page](#) [Int](#) , [size](#) [Int](#)) [ImagePage!](#)

Mutation

Mutations

FIELDS

⌵ None

- `registerUser` (`input` `CreateUserInput!`) `NewUser!`
Authentication and User Management
- `resetPassword` (`input` `ResetPasswordInput!`) `AuthenticateResponse!`
- `authenticateByBasic` (`username` `String!` , `password` `String!`) `AuthenticateResponse!`
- `updateUserProfile` (`input` `UpdateUserProfileInput!`) `UserProfile!`
- `createPost` (`input` `CreatePostInput!`) `Post!`
Posts and Comments
- `updatePost` (`postId` `ID!` , `input` `UpdatePostInput!`) `Post!`
- `deletePost` (`postId` `ID!`) `Boolean!`
- `addReactionToPost` (`postId` `ID!` , `input` `CreateReactionInput!`) `[ReactionList!]!`
- `createComment` (`postId` `ID!` , `input` `CreateCommentInput!`) `Comment!`
- `updateComment` (`postId` `ID!` , `commentId` `ID!` , `input` `UpdateCommentInput!`) `Comment!`
- `deleteComment` (`postId` `ID!` , `commentId` `ID!`) `Boolean!`
- `addReactionToComment` (`postId` `ID!` , `commentId` `ID!` , `input` `CreateReactionInput!`) `[ReactionList!]!`
- `createNestedComment` (`postId` `ID!` , `commentId` `ID!` , `input` `CreateCommentInput!`) `Comment!`
- `createMessage` (`userId` `ID!` , `input` `CreateMessageInput!`) `Message!`
Messages
- `updateMessage` (`userId` `ID!` , `messageId` `ID!` , `input` `UpdateMessageInput!`) `Message!`
- `deleteMessage` (`userId` `ID!` , `messageId` `ID!`) `Boolean!`
- `addReactionToMessage` (`userId` `ID!` , `messageId` `ID!` , `input` `CreateReactionInput!`) `[ReactionList!]!`

- `addReactionToMessageImage (userId ID! , messageId ID! , imageId ID! , input CreateReactionInput!)` `[ReactionList]!`
- `followUser (userId ID!)` `Boolean!`
Relationships
- `unfollowUser (userId ID!)` `Boolean!`
- `friendUser (userId ID!)` `Boolean!`
- `unfriendUser (userId ID!)` `Boolean!`
- `createGroup (input CreateGroupInput!)` `Participant!`
Groups
- `joinGroup (groupId ID!)` `Boolean!`
- `leaveGroup (groupId ID!)` `Boolean!`
- `createGroupPost (groupId ID! , input CreatePostInput!)` `Post!`
- `acceptGroupJoinRequest (groupId ID! , requestId ID!)` `Boolean!`
- `rejectGroupJoinRequest (groupId ID! , requestId ID!)` `Boolean!`
- `acceptUserRequest (requestId ID!)` `Boolean!`
User Requests
- `rejectUserRequest (requestId ID!)` `Boolean!`
- `uploadImage` `Image!`
Images

types.graphqls

```

1  # Enums
2  enum RequestType {
3      FOLLOW
4      FRIEND
5      JOIN
6  }
7
8  type User {
9      id: ID!
10     name: String!
11     avatarUrl: String
12     isGroup: Boolean!
13     profile: UserProfile!
14     friends(page: Int, size: Int): UserPage!
15     followers(page: Int, size: Int): UserPage!
16     requests(page: Int, size: Int): RequestPage!
17     feed(page: Int, size: Int): PostPage!
18     post(postId: ID!): Post
19     posts(page: Int, size: Int): PostPage!
20 }
21
22 # Participant Types
23 type Participant {
24     id: ID!
25     name: String!
26     avatarUrl: String
27     isGroup: Boolean!
28 }
29
30 # Reaction Types
31 type ReactionList {
32     reactionValue: Int!
33     reactionCount: Int!
34     participants: [User!]!
35 }

```

```

36
37 # Image Type
38 type Image {
39     id: ID!
40     uri: String!
41     reactions: [ReactionList!]!
42 }
43
44 # Message Types
45 type Message {
46     id: ID!
47     author: Participant!
48     text: String!
49     images: [Image!]!
50     reactions: [ReactionList!]!
51 }
52
53 input CreateMessageInput {
54     text: String!
55     images: [ID!]!
56 }
57
58 input UpdateMessageInput {
59     text: String!
60     images: [ID!]!
61 }
62
63 # Dialog Info Type
64 type DialogInfo {
65     opponent: Participant!
66     newMessageCount: Int!
67 }

```

```

68
69 # Post Types
70 type Post {
71     id: ID!
72     author: User!
73     text: String!
74     images: [Image!]!
75     reactions: [ReactionList!]!
76     comments(page: Int, size: Int): CommentPage!
77 }
78
79 input CreatePostInput {
80     text: String!
81     images: [ID!]!
82 }
83
84 input UpdatePostInput {
85     text: String!
86     images: [ID!]!
87 }
88
89 # Comment Types
90 type Comment {
91     id: ID!
92     postId: ID!
93     parentCommentId: ID
94     author: Participant!
95     text: String!
96     images: [Image!]!
97     reactions: [ReactionList!]!
98     comments(page: Int, size: Int): CommentPage!
99 }
100
101 input CreateCommentInput {
102     text: String!
103     images: [ID!]!
104 }
105
106 input UpdateCommentInput {
107     text: String!
108     images: [ID!]!
109 }
110
111 # Reaction Input
112 input CreateReactionInput {
113     value: Int!
114 }
115
116 # User Profile Types
117 type UserProfile {
118     name: String!
119     handle: String!
120     profileImage: Image
121     coverPhoto: Image
122     phoneNumber: String
123     bio: String
124     country: String
125     city: String
126 }
127
128 input UpdateUserProfileInput {
129     name: String
130     handle: String
131     profileImageId: ID
132     coverPhotoImageId: ID
133     phoneNumber: String
134     bio: String
135     country: String
136     city: String
137 }
138

```

```

139 # Group Profile Types
140 type GroupProfile {
141     name: String!
142     handle: String!
143     profileImage: Image
144     coverPhoto: Image
145 }
146
147 input CreateGroupInput {
148     name: String!
149     isPrivate: Boolean
150 }
151
152 # Request Types
153 type Request {
154     id: ID!
155     sender: User!
156     type: RequestType!
157 }
158
159 # Pagination Types
160 type PageInfo {
161     totalElements: Int!
162     totalPages: Int!
163     pageNumber: Int!
164     pageSize: Int!
165     first: Boolean!
166     last: Boolean!
167 }
168
169 type PostPage {
170     content: [Post!]!
171     pageInfo: PageInfo!
172 }
173

```

```

174 type CommentPage {
175     content: [Comment!]!
176     pageInfo: PageInfo!
177 }
178
179 type UserPage {
180     content: [User!]!
181     pageInfo: PageInfo!
182 }
183
184 type ImagePage {
185     content: [Image!]!
186     pageInfo: PageInfo!
187 }
188
189 type MessagePage {
190     content: [Message!]!
191     pageInfo: PageInfo!
192 }
193
194 type DialogInfoPage {
195     content: [DialogInfo!]!
196     pageInfo: PageInfo!
197 }
198
199 type RequestPage {
200     content: [Request!]!
201     pageInfo: PageInfo!
202 }
203
204 # Authentication Types
205 type AuthenticateResponse {
206     token: String!
207 }
208

```

```

209 type NewUser {
210     id: ID!
211     username: String!
212     fullName: String!
213     avatarUrl: String!
214     passwordResetId: String!
215 }
216
217 input CreateUserInput {
218     email: String!
219     fullName: String!
220 }
221
222 input AuthenticateInput {
223     email: String
224     username: String
225     password: String
226     token: String
227 }
228
229 input ResetPasswordInput {
230     passwordResetToken: String!
231     password: String!
232 }
233

```