**/v1**

/authentication/uid/signedInStatus

**POST:**

request: { signIn: <bool>, deviceToken: <str> (opt.) }

response: { isSignedIn: <bool>, user: <User> (opt.) }

/authentication/deviceSignedInOn/?deviceToken=<str>

**GET:**

response: { isSignedIn: <bool>, user: <User> (opt.) }

/chats/uid

**GET:**

response: { chats: <List<Chat>> }

**POST:**

request: { chatName: <str> }

response: status code

/chats/uid/chatID

**POST:**

request: {

isPost: <bool>,

isImage: <bool> (opt.),

text: <str> (opt.)

}, FILE (opt.)

response: status code

**DELETE:**

response: status code

/chats/uid/chatID/members

**GET:**

request: { members: <List<User>> }

**POST:**

request: { uid: <str> }

response: status code

**DELETE:**

request: { uid: <str> }

response: status code

/comments/postID

**GET:**

response: { comments: <List<Comment>> }

**POST:**

request: { path: <str>, comment: <str>, uid: <str> }

response: { reasonForDenial: <str> (opt.) }

/comments/postID/report

**POST:**

request: { comment: <Comment> }

response: status code

new\_content/uid/recommendations/

**GET:**

response: { posts: <List<Post>> }

new\_content/uid/following/

**GET:**

response: { posts: <List<Post>> }

/posts/postID/watched\_list

**POST:**

request: { uid: <str>, userRating: <int> }

response: status code

/posts/uid

**GET:**

response: { posts: <List<Post>> }

**POST:**

request: { isImage: <bool>, isPrivate: <bool> }, FILE

response: status code

**POST:**

request: { reportType: <int> }

response: status code

/posts/uid/profile

**GET:**

response: { <Profile> }

**POST:**

request: { isImage: <bool> }, FILE

response: status code

/posts/uid/postID

**POST:**

request: { isPrivate: <bool> }

response: status code

**DELETE:**

response: status code

/posts/uid/postID/reports

**POST:**

request: null

response: status code

/relationships/uid/following/

**GET:**

response: { following: <List<(user: <User>, isFriends: <bool>)> }

**POST:**

request: { uid: <str> }

response: status code

/relationships/uid/blocked/

**POST:**

request: { uid: <str> }

response = status code

/relationships/uid/following/uid/

**GET:**

response: { isFollowing: <bool> }

**POST:**

request: { followBack: <bool> }

response: status code

**DELETE:**

response: status code

/relationships/uid/followers/

**GET:**

response: { following: <List<User>> }

/relationships/uid/followers/new/

**GET:**

response: { followerList: <List<User>> }

/relationships/uid/friends/

**GET:**

response: { friends: <List<User>> }

/users/?contains=‘searchString’

**GET:**

response: { creatorsList: <List<User>> }

/users/preferences

**GET:**

response: { fields: <List<str>> }

/users/new

**POST:**

request:

{

uid: <str>,

userID: <str>,

username: <str>,

email: <str>,

phone: <str>

}

response:

statusCode: 201

{

user: <User>

}

statusCode: 200,

{

fieldsTaken: <List<str>>

}

/users/uid

**GET:**

response: { user: <User> }

**POST:**

request: { profileColor: <str>, username: <str> }

response:

statusCode: 200

**DELETE:**

response: status code

/users/uid/preferences

**POST:**

request: { preferences: <List<bool>> }

response: status code

**Models**:

Profile:

{

exists: <bool>,

isImage: <bool>,

downloadURL: <str>

}

User:

{

uid: <str>,

userID: <str>,

username: <str>,

profileColor: <str>

}

Chat:

{

chatID: <str>,

chatname: <str>,

isDirectMessage: <bool>,

Members: <List<User>>

}

ChatItem

{

isPost: True,

uid: <str>

date: <datetime>

post:

{

isImage: <bool>,

downloadURL: <str>,

} (opt.)

text: <str> (opt.)

}

Post:

{

creator: <User>,

postID: <str>,

isImage: <bool>,

downloadURL: <str>,

}

Comment:

{

datePosted: <datetime>

uid: <str>

comment: <str>

path: <str>

}