

---

# **Amino.py**

***Release 2.0.0***

**Aug 18, 2021**



---

## Contents:

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Installing . . . . .	1
<b>2</b>	<b>API</b>	<b>3</b>
2.1	Modules . . . . .	3
2.2	Objects . . . . .	36
2.3	Exceptions . . . . .	41
2.4	Device . . . . .	53
2.5	Headers . . . . .	53
2.6	Helpers . . . . .	53
<b>3</b>	<b>Contact</b>	<b>55</b>
<b>4</b>	<b>Indices and tables</b>	<b>57</b>
	<b>Python Module Index</b>	<b>59</b>
	<b>Index</b>	<b>61</b>



# CHAPTER 1

---

## Introduction

---

This is the documentation for Amino.py, an library to make Amino bots.

### 1.1 Installing

You can get the library directly from PyPI:

```
python3 -m pip install -U amino.py
```

If you are using Windows, then the following should be used instead:

```
py -3 -m pip install -U amino.py
```

And from the source code, make sure you're on the same directory as the `setup.py` file:

```
python3 setup.py install
```



## 2.1 Modules

### 2.1.1 ACM

```
class amino.acm.ACM(profile: amino.lib.util.objects.UserProfile, comId: str = None)
    Bases: amino.client.Client

    accept_join_request (userId: str)
    add_influencer (userId: str, monthlyFee: int)
    change_guidelines (message: str)
    change_module (module: str, isEnabled: bool)
    change_sidepanel_color (color: str)
    change_welcome_message (message: str, isEnabled: bool = True)
    create_community (name: str, tagline: str, icon: BinaryIO, themeColor: str, joinType: int = 0,
        primaryLanguage: str = 'en')
    delete_community (email: str, password: str, verificationCode: str)
    delete_pending_role (noticeId: str)
    edit_community (name: str = None, description: str = None, aminoId: str = None, primaryLanguage: str = None, themePackUrl: str = None)
    get_categories (start: int = 0, size: int = 25)
    get_community_stats ()
    get_community_user_stats (type: str, start: int = 0, size: int = 25)
    get_join_requests (start: int = 0, size: int = 25)
    get_notice_list (start: int = 0, size: int = 25)
```

```
list_communities (start: int = 0, size: int = 25)
parse_headers (data=None)
promote (userId: str, rank: str)
reject_join_request (userId: str)
remove_influencer (userId: str)
upload_themepack_raw (file: BinaryIO)
```

## 2.1.2 Client

```
class amino.client.Client (deviceId: str = None, socketDebugging=False)
    Bases: amino.socket.Callbacks, amino.socket.SocketHandler
```

```
accept_host (chatId: str, requestId: str)
accept_organizer (chatId: str, requestId: str)
activate_account (email: str, code: str)
```

Activate an account.

### Parameters

- **email** : Email of the account.
- **code** : Verification code.

### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

```
add_linked_community (comId: str)
    Add a Linked Community on your profile.
```

### Parameters

- **comId** : ID of the Community.

### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

```
block (userId: str)
    Block an User.
```

### Parameters

- **userId** : ID of the User.

### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

```
change_password (email: str, password: str, code: str)
    Change password of an account.
```

### Parameters

- **email** : Email of the account.



- **password** : Password of the account.
- **code** : Verification code.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**check\_device** (*deviceId: str*)

Check if the Device ID is valid.

**Parameters**

- **deviceId** : ID of the Device.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**claim\_new\_user\_coupon** ()

Claim the New User Coupon available when a new account is created.

**Parameters**

- No parameters required.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**comment** (*message: str, userId: str = None, blogId: str = None, wikiId: str = None, replyTo: str = None*)

Comment on a User's Wall, Blog or Wiki.

**Parameters**

- **message** : Message to be sent.
- **userId** : ID of the User. (for Walls)
- **blogId** : ID of the Blog. (for Blogs)
- **wikiId** : ID of the Wiki. (for Wikis)
- **replyTo** : ID of the Comment to Reply to.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**configure** (*age: int, gender: str*)

Configure the settings of an account.

**Parameters**

- **age** : Age of the account. Minimum is 13.
- **gender** [Gender of the account.]
  - Male, Female or Non-Binary

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**delete\_account** (*password: str*)

Delete an account.

**Parameters**

- **password** : Password of the account.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**delete\_comment** (*commentId: str, userId: str = None, blogId: str = None, wikiId: str = None*)

Delete a Comment on a User's Wall, Blog or Wiki.

**Parameters**

- **commentId** : ID of the Comment.
- **userId** : ID of the User. (for Walls)
- **blogId** : ID of the Blog. (for Blogs)
- **wikiId** : ID of the Wiki. (for Wikis)

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**delete\_message** (*chatId: str, messageId: str, asStaff: bool = False, reason: str = None*)

Delete a Message from a Chat.

**Parameters**

- **messageId** : ID of the Message.
- **chatId** : ID of the Chat.
- **asStaff** : If execute as a Staff member (Leader or Curator).
- **reason** : Reason of the action to show on the Moderation History.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**edit\_chat** (*chatId: str, doNotDisturb: bool = None, pinChat: bool = None, title: str = None, icon: str = None, backgroundImage: BinaryIO = None, content: str = None, announcement: str = None, coHosts: list = None, keywords: list = None, pinAnnouncement: bool = None, publishToGlobal: bool = None, canTip: bool = None, viewOnly: bool = None, canInvite: bool = None, fansOnly: bool = None*)

Send a Message to a Chat.

**Parameters**

- **chatId** : ID of the Chat.
- **title** : Title of the Chat.
- **content** : Content of the Chat.

- **icon** : Icon of the Chat.
- **backgroundImage** : Background Image of the Chat.
- **announcement** : Announcement of the Chat.
- **pinAnnouncement** : If the Chat Announcement should Pinned or not.
- **coHosts** : List of User IDS to be Co-Host.
- **keywords** : List of Keywords of the Chat.
- **viewOnly** : If the Chat should be on View Only or not.
- **canTip** : If the Chat should be Tippable or not.
- **canInvite** : If the Chat should be Invitable or not.
- **fansOnly** : If the Chat should be Fans Only or not.
- **publishToGlobal** : If the Chat should show on Public Chats or not.
- **doNotDisturb** : If the Chat should Do Not Disturb or not.
- **pinChat** : If the Chat should Pinned or not.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**edit\_profile** (*nickname: str = None, content: str = None, icon: BinaryIO = None, backgroundColor: str = None, backgroundImage: str = None, defaultBubbleId: str = None*)  
 Edit account's Profile.

#### Parameters

- **nickname** : Nickname of the Profile.
- **content** : Biography of the Profile.
- **icon** : Icon of the Profile.
- **backgroundImage** : Url of the Background Picture of the Profile.
- **backgroundColor** : Hexadecimal Background Color of the Profile.
- **defaultBubbleId** : Chat bubble ID.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**end\_vc** (*comId: str, chatId: str, joinType: int = 2*)

**flag** (*reason: str, flagType: int, userId: str = None, blogId: str = None, wikiId: str = None, asGuest: bool = False*)  
 Flag a User, Blog or Wiki.

#### Parameters

- **reason** : Reason of the Flag.
- **flagType** : Type of the Flag.
- **userId** : ID of the User.
- **blogId** : ID of the Blog.

- **wikiId** : ID of the Wiki.
- *asGuest* : Execute as a Guest.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**flag\_community** (*comId: str, reason: str, flagType: int, isGuest: bool = False*)  
Flag a Community.

**Parameters**

- **comId** : ID of the Community.
- **reason** : Reason of the Flag.
- **flagType** : Type of Flag.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**follow** (*userId: Union[str, list]*)  
Follow an User or Multiple Users.

**Parameters**

- **userId** : ID of the User or List of IDs of the Users.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**get\_account\_info** ()

**get\_all\_users** (*start: int = 0, size: int = 25*)  
Get list of users of Amino.

**Parameters**

- *start* : Where to start the list.
- *size* : Size of the list.

**Returns**

- **Success** : *User Profile Count List Object*
- **Fail** : *Exceptions*

**get\_avatar\_frames** (*start: int = 0, size: int = 25*)

**get\_blocked\_users** (*start: int = 0, size: int = 25*)  
List of Users that the User Blocked.

**Parameters**

- *start* : Where to start the list.
- *size* : Size of the list.

**Returns**

- **Success** : *Users List*
- **Fail** : *Exceptions*

**get\_blocker\_users** (*start: int = 0, size: int = 25*)

List of Users that are Blocking the User.

#### Parameters

- *start* : Where to start the list.
- *size* : Size of the list.

#### Returns

- **Success** : List of User IDs
- **Fail** : *Exceptions*

**get\_blog\_comments** (*blogId: str = None, wikiId: str = None, quizId: str = None, fileId: str = None, sorting: str = 'newest', start: int = 0, size: int = 25*)

**get\_blog\_info** (*blogId: str = None, wikiId: str = None, quizId: str = None, fileId: str = None*)

**get\_chat\_messages** (*chatId: str, size: int = 25, pageToken: str = None*)

List of Messages from an Chat.

#### Parameters

- **chatId** : ID of the Chat.
- *size* : Size of the list.
- *size* : Size of the list.
- *pageToken* : Next Page Token.

#### Returns

- **Success** : *Message List*
- **Fail** : *Exceptions*

**get\_chat\_thread** (*chatId: str*)

Get the Chat Object from an Chat ID.

#### Parameters

- **chatId** : ID of the Chat.

#### Returns

- **Success** : *Chat Object*
- **Fail** : *Exceptions*

**get\_chat\_threads** (*start: int = 0, size: int = 25*)

List of Chats the account is in.

#### Parameters

- *start* : Where to start the list.
- *size* : Size of the list.

#### Returns

- **Success** : *Chat List*
- **Fail** : *Exceptions*

**get\_chat\_users** (*chatId: str, start: int = 0, size: int = 25*)

**get\_community\_info** (*comId: str*)

Information of an Community.

**Parameters**

- **comId** : ID of the Community.

**Returns**

- **Success** : *Community Object*
- **Fail** : *Exceptions*

**get\_eventlog** (*language: str = 'en'*)

**get\_from\_code** (*code: str*)

Get the Object Information from the Amino URL Code.

**Parameters**

- **code** : Code from the Amino URL.

**Returns**

- **Success** : *From Code Object*
- **Fail** : *Exceptions*

**get\_from\_deviceid** (*deviceId: str*)

Get the User ID from an Device ID.

**Parameters**

- **deviceId** : ID of the Device.

**Returns**

- **Success** : User ID
- **Fail** : *Exceptions*

**get\_from\_id** (*objectId: str, objectType: int, comId: str = None*)

Get the Object Information from the Object ID and Type.

**Parameters**

- **objectId** : ID of the Object. User ID, Blog ID, etc.
- **objectType** : Type of the Object.
- **comId** : ID of the Community. Use if the Object is in a Community.

**Returns**

- **Success** : *From Code Object*
- **Fail** : *Exceptions*

**get\_linked\_communities** (*userId: str*)

Get a List of Linked Communities of an User.

**Parameters**

- **userId** : ID of the User.

**Returns**

- **Success** : *Community List*

- **Fail** : *Exceptions*

**get\_membership\_info()**

Get Information about your Amino+ Membership.

**Parameters**

- No parameters required.

**Returns**

- **Success** : *Membership Object*
- **Fail** : *Exceptions*

**get\_message\_info(chatId: str, messageId: str)**

Information of an Message from an Chat.

**Parameters**

- **chatId** : ID of the Chat.
- **messageId** : ID of the Message.

**Returns**

- **Success** : *Message Object*
- **Fail** : *Exceptions*

**get\_subscriptions(start: int = 0, size: int = 25)**

Get Information about the account's Subscriptions.

**Parameters**

- **start** : Where to start the list.
- **size** : Size of the list.

**Returns**

- **Success** : *List*
- **Fail** : *Exceptions*

**get\_supported\_languages()**

Get the List of Supported Languages by Amino.

**Parameters**

- No parameters required.

**Returns**

- **Success** : *List of Supported Languages*
- **Fail** : *Exceptions*

**get\_ta\_announcements(language: str = 'en', start: int = 0, size: int = 25)**

Get the list of Team Amino's Announcement Blogs.

**Parameters**

- **language** [Language of the Blogs.]
  - en, es, pt, ar, ru, fr, de
- **start** : Where to start the list.
- **size** : Size of the list.

**Returns**

- **Success** : *Blogs List*
- **Fail** : *Exceptions*

**get\_unlinked\_communities** (*userId: str*)

Get a List of Unlinked Communities of an User.

**Parameters**

- **userId** : ID of the User.

**Returns**

- **Success** : *Community List*
- **Fail** : *Exceptions*

**get\_user\_followers** (*userId: str, start: int = 0, size: int = 25*)

List of Users that are Following the User.

**Parameters**

- **userId** : ID of the User.
- **start** : Where to start the list.
- **size** : Size of the list.

**Returns**

- **Success** : *User List*
- **Fail** : *Exceptions*

**get\_user\_following** (*userId: str, start: int = 0, size: int = 25*)

List of Users that the User is Following.

**Parameters**

- **userId** : ID of the User.
- **start** : Where to start the list.
- **size** : Size of the list.

**Returns**

- **Success** : *User List*
- **Fail** : *Exceptions*

**get\_user\_info** (*userId: str*)

Information of an User.

**Parameters**

- **userId** : ID of the User.

**Returns**

- **Success** : *User Object*
- **Fail** : *Exceptions*

**get\_user\_visitors** (*userId: str, start: int = 0, size: int = 25*)

List of Users that Visited the User.

**Parameters**



- **userId** : ID of the User.
- **start** : Where to start the list.
- **size** : Size of the list.

**Returns**

- **Success** : *Visitors List*
- **Fail** : *Exceptions*

**get\_wall\_comments** (*userId: str, sorting: str, start: int = 0, size: int = 25*)  
List of Wall Comments of an User.

**Parameters**

- **userId** : ID of the User.
- **sorting** [Order of the Comments.]
  - newest, oldest, top
- **start** : Where to start the list.
- **size** : Size of the list.

**Returns**

- **Success** : *Comments List*
- **Fail** : *Exceptions*

**get\_wallet\_history** (*start: int = 0, size: int = 25*)  
Get the Wallet's History Information.

**Parameters**

- **start** : Where to start the list.
- **size** : Size of the list.

**Returns**

- **Success** : *Wallet Object*
- **Fail** : *Exceptions*

**get\_wallet\_info** ()  
Get Information about the account's Wallet.

**Parameters**

- No parameters required.

**Returns**

- **Success** : *Wallet Object*
- **Fail** : *Exceptions*

**handle\_socket\_message** (*data*)

**invite\_to\_chat** (*userId: Union[str, list], chatId: str*)  
Invite a User or List of Users to a Chat.

**Parameters**

- **userId** : ID of the User or List of User IDs.

- **chatId** : ID of the Chat.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**invite\_to\_vc** (*chatId: str, userId: str*)

Invite a User to a Voice Chat

**Parameters**

- **chatId** - ID of the Chat
- **userId** - ID of the User

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**join\_chat** (*chatId: str*)

Join an Chat.

**Parameters**

- **chatId** : ID of the Chat.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**join\_community** (*comId: str, invitationCode: str = None*)

Join a Community.

**Parameters**

- **comId** : ID of the Community.
- **invitationCode** : Invitation Code.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**join\_video\_chat** (*comId: str, chatId: str, joinType: int = 1*)

Joins a Video Chat

**Parameters**

- **comId** : ID of the Community
- **chatId** : ID of the Chat

**join\_video\_chat\_as\_viewer** (*comId: str, chatId: str*)

**join\_voice\_chat** (*comId: str, chatId: str, joinType: int = 1*)

Joins a Voice Chat

**Parameters**

- **comId** : ID of the Community
- **chatId** : ID of the Chat

**kick** (*userId: str, chatId: str, allowRejoin: bool = True*)

**leave\_chat** (*chatId: str*)

Leave an Chat.

#### Parameters

- **chatId** : ID of the Chat.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**leave\_community** (*comId: str*)

Leave a Community.

#### Parameters

- **comId** : ID of the Community.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**like\_blog** (*blogId: Union[str, list] = None, wikiId: str = None*)

Like a Blog, Multiple Blogs or a Wiki.

#### Parameters

- **blogId** : ID of the Blog or List of IDs of the Blogs. (for Blogs)
- **wikiId** : ID of the Wiki. (for Wikis)

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**like\_comment** (*commentId: str, userId: str = None, blogId: str = None, wikiId: str = None*)

Like a Comment on a User's Wall, Blog or Wiki.

#### Parameters

- **commentId** : ID of the Comment.
- **userId** : ID of the User. (for Walls)
- **blogId** : ID of the Blog. (for Blogs)
- **wikiId** : ID of the Wiki. (for Wikis)

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**link\_identify** (*code: str*)

**login** (*email: str, password: str*)

Login into an account.

#### Parameters

- **email** : Email of the account.

- **password** : Password of the account.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**login\_sid** (*SID: str*)

Login into an account with an SID

**Parameters**

- **SID** : SID of the account

**logout** ()

Logout from an account.

**Parameters**

- No parameters required.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**mark\_as\_read** (*chatId: str, messageId: str*)

Mark a Message from a Chat as Read.

**Parameters**

- **messageId** : ID of the Message.
- **chatId** : ID of the Chat.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**parse\_headers** (*data=None*)**register** (*nickname: str, email: str, password: str, verificationCode: str, deviceId: str = '01B592EF5658F82E1339B39AA893FF661D7E8B8F1D16227E396EF4B1BF60F33D25566A35AB1514DAB5'*)  
Register an account.**Parameters**

- **nickname** : Nickname of the account.
- **email** : Email of the account.
- **password** : Password of the account.
- **verificationCode** : Verification code.
- **deviceId** : The device id being registered to.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**remove\_linked\_community** (*comId: str*)

Remove a Linked Community on your profile.

**Parameters**

- **comId** : ID of the Community.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**reorder\_linked\_communities** (*comIds: list*)

Reorder List of Linked Communities.

**Parameters**

- **comIds** : IDS of the Communities.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**request\_join\_community** (*comId: str, message: str = None*)

Request to join a Community.

**Parameters**

- **comId** : ID of the Community.
- **message** : Message to be sent.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**request\_verify\_code** (*email: str, resetPassword: bool = False*)

Request an verification code to the targeted email.

**Parameters**

- **email** : Email of the account.
- **resetPassword** : If the code should be for Password Reset.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**restore** (*email: str, password: str*)

Restore a deleted account.

**Parameters**

- **email** : Email of the account.
- **password** : Password of the account.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**run\_vc** (*comId: str, chatId: str, joinType: str*)

**search\_community** (*aminoId: str*)

Search a Community by its Amino ID.

**Parameters**

- **aminoId** : Amino ID of the Community.

**Returns**

- **Success** : *Community List*
- **Fail** : *Exceptions*

**send\_coins** (*coins: int, blogId: str = None, chatId: str = None, objectId: str = None, transactionId: str = None*)

**send\_message** (*chatId: str, message: str = None, messageType: int = 0, file: BinaryIO = None, fileType: str = None, replyTo: str = None, mentionUserIds: list = None, stickerId: str = None, embedId: str = None, embedType: int = None, embedLink: str = None, embedTitle: str = None, embedContent: str = None, embedImage: BinaryIO = None*)

Send a Message to a Chat.

**Parameters**

- **message** : Message to be sent
- **chatId** : ID of the Chat.
- **file** : File to be sent.
- **fileType** [Type of the file.]
  - audio, image, gif
- **messageType** : Type of the Message.
- **mentionUserIds** : List of User IDS to mention. '@' needed in the Message.
- **replyTo** : Message ID to reply to.
- **stickerId** : Sticker ID to be sent.
- **embedTitle** : Title of the Embed.
- **embedContent** : Content of the Embed.
- **embedLink** : Link of the Embed.
- **embedImage** : Image of the Embed.
- **embedId** : ID of the Embed.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**set\_amino\_id** (*aminoId: str*)

Edit account's Amino ID.

**Parameters**

- **aminoId** : Amino ID of the Account.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**set\_privacy\_status** (*isAnonymous: bool = False, getNotifications: bool = False*)  
 Edit account's Privacy Status.

#### Parameters

- **isAnonymous** : If visibility should be Anonymous or not.
- **getNotifications** : If account should get new Visitors Notifications.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**start\_chat** (*userId: Union[str, list], message: str, title: str = None, content: str = None, isGlobal: bool = False, publishToGlobal: bool = False*)  
 Start an Chat with an User or List of Users.

#### Parameters

- **userId** : ID of the User or List of User IDs.
- **message** : Starting Message.
- **title** : Title of Group Chat.
- **content** : Content of Group Chat.
- **isGlobal** : If Group Chat is Global.
- **publishToGlobal** : If Group Chat should show in Global.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**start\_vc** (*comId: str, chatId: str, joinType: int = 1*)

**sub\_clients** (*start: int = 0, size: int = 25*)  
 List of Communities the account is in.

#### Parameters

- *start* : Where to start the list.
- *size* : Size of the list.

#### Returns

- **Success** : *Community List*
- **Fail** : *Exceptions*

**sub\_clients\_profile** (*start: int = 0, size: int = 25*)

**subscribe\_amino\_plus** (*transactionId="", sku='d940cf4a-6cf2-4737-9f3d-655234a92ea5'*)  
 Subscribes to amino+

#### Parameters

- **transactionId** - The transaction Id as a uuid4

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**unblock** (*userId: str*)

Unblock an User.

**Parameters**

- **userId** : ID of the User.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**unfollow** (*userId: str*)

Unfollow an User.

**Parameters**

- **userId** : ID of the User.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**unlike\_blog** (*blogId: str = None, wikiId: str = None*)

Remove a like from a Blog or Wiki.

**Parameters**

- **blogId** : ID of the Blog. (for Blogs)
- **wikiId** : ID of the Wiki. (for Wikis)

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**unlike\_comment** (*commentId: str, userId: str = None, blogId: str = None, wikiId: str = None*)

Remove a like from a Comment on a User's Wall, Blog or Wiki.

**Parameters**

- **commentId** : ID of the Comment.
- **userId** : ID of the User. (for Walls)
- **blogId** : ID of the Blog. (for Blogs)
- **wikiId** : ID of the Wiki. (for Wikis)

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**upload\_media** (*file: BinaryIO, fileType: str*)

Upload file to the amino servers.

**Parameters**

- **file** : File to be uploaded.

**Returns**

- **Success** : Url of the file uploaded to the server.



- **Fail** : *Exceptions*

**verify** (*email: str, code: str*)  
Verify an account.

**Parameters**

- **email** : Email of the account.
- **code** : Verification code.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**visit** (*userId: str*)  
Visit an User.

**Parameters**

- **userId** : ID of the User.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**wallet\_config** (*level: int*)  
Changes ads config

**Parameters**

- **level - Level of the ads.**  
– 1, 2

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

### 2.1.3 SubClient

**class** amino.sub\_client.**SubClient** (*comId: str = None, aminoId: str = None, \*, profile: amino.lib.util.objects.UserProfile*)

Bases: *amino.client.Client*

**accept\_host** (*chatId: str, requestId: str*)

**accept\_organizer** (*chatId: str, requestId: str*)

**accept\_wiki\_request** (*requestId: str, destinationCategoryIdList: list*)

**activate\_bubble** (*bubbleId: str*)

**activity\_status** (*status: str*)

**add\_poll\_option** (*blogId: str, question: str*)

**add\_to\_favorites** (*userId: str*)

**apply\_avatar\_frame** (*avatarId: str, applyToAll: bool = True*)  
Apply avatar frame.

**Parameters**

- **avatarId** : ID of the avatar frame.
- **applyToAll** : Apply to all.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**apply\_bubble** (*bubbleId: str, chatId: str, applyToAll: bool = False*)

**ban** (*userId: str, reason: str, banType: int = None*)

**block** (*userId: str*)

Block an User.

**Parameters**

- **userId** : ID of the User.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**check\_in** (*tz: int = 0*)

**check\_notifications** ()

**claim\_vc\_reputation** (*chatId: str*)

**clear\_notifications** ()

**comment** (*message: str, userId: str = None, blogId: str = None, wikiId: str = None, replyTo: str = None, isGuest: bool = False*)

Comment on a User's Wall, Blog or Wiki.

**Parameters**

- **message** : Message to be sent.
- **userId** : ID of the User. (for Walls)
- **blogId** : ID of the Blog. (for Blogs)
- **wikiId** : ID of the Wiki. (for Wikis)
- **replyTo** : ID of the Comment to Reply to.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**create\_shared\_folder** (*title: str*)

**create\_wiki\_category** (*title: str, parentCategoryId: str, content: str = None*)

**deactivate\_bubble** (*bubbleId: str*)

**delete\_blog** (*blogId: str*)

**delete\_chat** (*chatId: str*)

Delete a Chat.

**Parameters**

- **chatId** : ID of the Chat.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**delete\_comment** (*commentId: str, userId: str = None, blogId: str = None, wikiId: str = None*)  
Delete a Comment on a User's Wall, Blog or Wiki.

#### Parameters

- **commentId** : ID of the Comment.
- **userId** : ID of the User. (for Walls)
- **blogId** : ID of the Blog. (for Blogs)
- **wikiId** : ID of the Wiki. (for Wikis)

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**delete\_invite\_code** (*inviteId: str*)

**delete\_message** (*chatId: str, messageId: str, asStaff: bool = False, reason: str = None*)  
Delete a Message from a Chat.

#### Parameters

- **messageId** : ID of the Message.
- **chatId** : ID of the Chat.
- **asStaff** : If execute as a Staff member (Leader or Curator).
- **reason** : Reason of the action to show on the Moderation History.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**delete\_notification** (*notificationId: str*)

**delete\_wiki** (*wikiId: str*)

**downvote\_comment** (*blogId: str, commentId: str*)

**edit\_blog** (*blogId: str, title: str = None, content: str = None, imageUrl: list = None, categoriesList: list = None, backgroundColor: str = None, fansOnly: bool = False*)

**edit\_chat** (*chatId: str, doNotDisturb: bool = None, pinChat: bool = None, title: str = None, icon: str = None, backgroundImage: BinaryIO = None, content: str = None, announcement: str = None, coHosts: list = None, keywords: list = None, pinAnnouncement: bool = None, publishToGlobal: bool = None, canTip: bool = None, viewOnly: bool = None, canInvite: bool = None, fansOnly: bool = None*)

Send a Message to a Chat.

#### Parameters

- **chatId** : ID of the Chat.
- **title** : Title of the Chat.

- **content** : Content of the Chat.
- **icon** : Icon of the Chat.
- **backgroundImage** : Background Image of the Chat.
- **announcement** : Announcement of the Chat.
- **pinAnnouncement** : If the Chat Announcement should Pinned or not.
- **coHosts** : List of User IDS to be Co-Host.
- **keywords** : List of Keywords of the Chat.
- **viewOnly** : If the Chat should be on View Only or not.
- **canTip** : If the Chat should be Tippable or not.
- **canInvite** : If the Chat should be Invitable or not.
- **fansOnly** : If the Chat should be Fans Only or not.
- **publishToGlobal** : If the Chat should show on Public Chats or not.
- **doNotDisturb** : If the Chat should Do Not Disturb or not.
- **pinChat** : If the Chat should Pinned or not.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**edit\_profile** (*nickname: str = None, content: str = None, icon: BinaryIO = None, chatRequestPrivilege: str = None, imageList: list = None, captionList: list = None, backgroundImage: str = None, backgroundColor: str = None, titles: list = None, colors: list = None, defaultBubbleId: str = None*)

Edit account's Profile.

#### Parameters

- **nickname** : Nickname of the Profile.
- **content** : Biography of the Profile.
- **icon** : Icon of the Profile.
- **backgroundImage** : Url of the Background Picture of the Profile.
- **backgroundColor** : Hexadecimal Background Color of the Profile.
- **defaultBubbleId** : Chat bubble ID.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**edit\_titles** (*userId: str, titles: list, colors: list*)

**feature** (*time: int, userId: str = None, chatId: str = None, blogId: str = None, wikiId: str = None*)

**flag** (*reason: str, flagType: int, userId: str = None, blogId: str = None, wikiId: str = None, asGuest: bool = False*)  
Flag a User, Blog or Wiki.

#### Parameters

- **reason** : Reason of the Flag.
- **flagType** : Type of the Flag.
- **userId** : ID of the User.
- **blogId** : ID of the Blog.
- **wikiId** : ID of the Wiki.
- *asGuest* : Execute as a Guest.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**follow** (*userId: Union[str, list]*)

Follow an User or Multiple Users.

**Parameters**

- **userId** : ID of the User or List of IDs of the Users.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**generate\_invite\_code** (*duration: int = 0, force: bool = True*)

**get\_all\_users** (*type: str = 'recent', start: int = 0, size: int = 25*)

Get list of users of Amino.

**Parameters**

- *start* : Where to start the list.
- *size* : Size of the list.

**Returns**

- **Success** : *User Profile Count List Object*
- **Fail** : *Exceptions*

**get\_best\_quiz** (*start: int = 0, size: int = 25*)

**get\_blocked\_users** (*start: int = 0, size: int = 25*)

List of Users that the User Blocked.

**Parameters**

- *start* : Where to start the list.
- *size* : Size of the list.

**Returns**

- **Success** : *Users List*
- **Fail** : *Exceptions*

**get\_blocker\_users** (*start: int = 0, size: int = 25*)

List of Users that are Blocking the User.

**Parameters**

- *start* : Where to start the list.
- *size* : Size of the list.

**Returns**

- **Success**: List of User IDs
- **Fail**: *Exceptions*

**get\_blog\_categories** (*size*: int = 25)

**get\_blog\_comments** (*blogId*: str = None, *wikiId*: str = None, *quizId*: str = None, *fileId*: str = None, *sorting*: str = 'newest', *start*: int = 0, *size*: int = 25)

**get\_blog\_info** (*blogId*: str = None, *wikiId*: str = None, *quizId*: str = None, *fileId*: str = None)

**get\_blog\_users** (*blogId*: str, *start*: int = 0, *size*: int = 25)

**get\_blogs\_by\_category** (*categoryId*: str, *start*: int = 0, *size*: int = 25)

**get\_bubble\_info** (*bubbleId*: str)

**get\_bubble\_template\_list** (*start*: int = 0, *size*: int = 25)

**get\_chat\_messages** (*chatId*: str, *size*: int = 25, *pageToken*: str = None)  
List of Messages from an Chat.

**Parameters**

- **chatId** : ID of the Chat.
- *size* : Size of the list.
- *pageToken* : Next Page Token.

**Returns**

- **Success**: *Message List*
- **Fail**: *Exceptions*

**get\_chat\_thread** (*chatId*: str)  
Get the Chat Object from an Chat ID.

**Parameters**

- **chatId** : ID of the Chat.

**Returns**

- **Success**: *Chat Object*
- **Fail**: *Exceptions*

**get\_chat\_threads** (*start*: int = 0, *size*: int = 25)  
List of Chats the account is in.

**Parameters**

- *start* : Where to start the list.
- *size* : Size of the list.

**Returns**

- **Success**: *Chat List*
- **Fail**: *Exceptions*

**get\_chat\_users** (*chatId*: str, *start*: int = 0, *size*: int = 25)

```

get_community_stickers ()
get_featured_users (start: int = 0, size: int = 25)
get_hidden_blogs (start: int = 0, size: int = 25)
get_influencer_fans (userId: str, start: int = 0, size: int = 25)
get_invite_codes (status: str = 'normal', start: int = 0, size: int = 25)
get_leaderboard_info (type: str, start: int = 0, size: int = 25)
get_live_layer ()
get_message_info (chatId: str, messageId: str)
    Information of an Message from an Chat.
    Parameters
        • chatId : ID of the Chat.
        • message : ID of the Message.
    Returns
        • Success : Message Object
        • Fail : Exceptions
get_notices (start: int = 0, size: int = 25)
    Parameters
        • start – Start of the List (Start: 0)
        • size – Amount of Notices to Show
    Returns Notices List
get_notifications (start: int = 0, size: int = 25)
get_online_favorite_users (start: int = 0, size: int = 25)
get_online_users (start: int = 0, size: int = 25)
get_public_chat_threads (type: str = 'recommended', start: int = 0, size: int = 25)
    List of Public Chats of the Community.
    Parameters
        • start : Where to start the list.
        • size : Size of the list.
    Returns
        • Success : Chat List
        • Fail : Exceptions
get_quiz_rankings (quizId: str, start: int = 0, size: int = 25)
get_recent_blogs (pageToken: str = None, start: int = 0, size: int = 25)
get_recent_quiz (start: int = 0, size: int = 25)
get_recent_wiki_items (start: int = 0, size: int = 25)
get_saved_blogs (start: int = 0, size: int = 25)
get_shared_folder_files (type: str = 'latest', start: int = 0, size: int = 25)

```

```
get_shared_folder_info()
get_sticker_collection(collectionId: str)
get_sticker_pack_info(sticker_pack_id: str)
get_sticker_packs()
get_store_chat_bubbles(start: int = 0, size: int = 25)
get_store_stickers(start: int = 0, size: int = 25)
get_tipped_users(blogId: str = None, wikiId: str = None, quizId: str = None, fileId: str = None,
                 chatId: str = None, start: int = 0, size: int = 25)
get_trending_quiz(start: int = 0, size: int = 25)
get_user_achievements(userId: str)
get_user_blogs(userId: str, start: int = 0, size: int = 25)
get_user_checkins(userId: str)
get_user_followers(userId: str, start: int = 0, size: int = 25)
    List of Users that are Following the User.
```

**Parameters**

- **userId** : ID of the User.
- **start** : Where to start the list.
- **size** : Size of the list.

**Returns**

- **Success** : *User List*
- **Fail** : *Exceptions*

```
get_user_following(userId: str, start: int = 0, size: int = 25)
    List of Users that the User is Following.
```

**Parameters**

- **userId** : ID of the User.
- **start** : Where to start the list.
- **size** : Size of the list.

**Returns**

- **Success** : *User List*
- **Fail** : *Exceptions*

```
get_user_info(userId: str)
    Information of an User.
```

**Parameters**

- **userId** : ID of the User.

**Returns**

- **Success** : *User Object*
- **Fail** : *Exceptions*



**get\_user\_visitors** (*userId: str, start: int = 0, size: int = 25*)

List of Users that Visited the User.

#### Parameters

- **userId** : ID of the User.
- **start** : Where to start the list.
- **size** : Size of the list.

#### Returns

- **Success** : *Visitors List*
- **Fail** : *Exceptions*

**get\_user\_wikis** (*userId: str, start: int = 0, size: int = 25*)

**get\_vc\_reputation\_info** (*chatId: str*)

**get\_wall\_comments** (*userId: str, sorting: str, start: int = 0, size: int = 25*)

List of Wall Comments of an User.

#### Parameters

- **userId** : ID of the User.
- **sorting** [Order of the Comments.]
  - newest, oldest, top
- **start** : Where to start the list.
- **size** : Size of the list.

#### Returns

- **Success** : *Comments List*
- **Fail** : *Exceptions*

**get\_wiki\_categories** (*start: int = 0, size: int = 25*)

**get\_wiki\_category** (*categoryId: str, start: int = 0, size: int = 25*)

**get\_wiki\_info** (*wikiId: str*)

**get\_wiki\_submissions** (*start: int = 0, size: int = 25*)

**hide** (*userId: str = None, chatId: str = None, blogId: str = None, wikiId: str = None, quizId: str = None, fileId: str = None, reason: str = None*)

**invite\_to\_chat** (*userId: Union[str, list], chatId: str*)

Invite a User or List of Users to a Chat.

#### Parameters

- **userId** : ID of the User or List of User IDs.
- **chatId** : ID of the Chat.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**invite\_to\_vc** (*chatId: str, userId: str*)

Invite a User to a Voice Chat

**Parameters**

- **chatId** - ID of the Chat
- **userId** - ID of the User

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**join\_chat** (*chatId: str*)

Join an Chat.

**Parameters**

- **chatId** : ID of the Chat.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**kick** (*userId: str, chatId: str, allowRejoin: bool = True*)**leave\_chat** (*chatId: str*)

Leave an Chat.

**Parameters**

- **chatId** : ID of the Chat.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**like\_blog** (*blogId: Union[str, list] = None, wikiId: str = None*)

Like a Blog, Multiple Blogs or a Wiki.

**Parameters**

- **blogId** : ID of the Blog or List of IDs of the Blogs. (for Blogs)
- **wikiId** : ID of the Wiki. (for Wikis)

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**like\_comment** (*commentId: str, userId: str = None, blogId: str = None, wikiId: str = None*)

Like a Comment on a User's Wall, Blog or Wiki.

**Parameters**

- **commentId** : ID of the Comment.
- **userId** : ID of the User. (for Walls)
- **blogId** : ID of the Blog. (for Blogs)
- **wikiId** : ID of the Wiki. (for Wikis)

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**lottery** (*tz*: int = 0)

**mark\_as\_read** (*chatId*: str, *messageId*: str)

Mark a Message from a Chat as Read.

#### Parameters

- **messageId** : ID of the Message.
- **chatId** : ID of the Chat.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**moderation\_history** (*userId*: str = None, *blogId*: str = None, *wikId*: str = None, *quizId*: str = None, *fileId*: str = None, *size*: int = 25)

**parse\_headers** (*data*=None)

**play\_quiz** (*quizId*: str, *questionIdsList*: list, *answerIdsList*: list, *quizMode*: int = 0)

**play\_quiz\_raw** (*quizId*: str, *quizAnswerList*: list, *quizMode*: int = 0)

**post\_blog** (*title*: str, *content*: str, *imageList*: list = None, *captionList*: list = None, *categoriesList*: list = None, *backgroundColor*: str = None, *fansOnly*: bool = False, *extensions*: dict = None)

**post\_wiki** (*title*: str, *content*: str, *icon*: str = None, *imageList*: list = None, *keywords*: str = None, *backgroundColor*: str = None, *fansOnly*: bool = False)

**promotion** (*noticeId*: str, *type*: str = 'accept')

**purchase** (*objectId*: str, *objectType*: int, *aminoPlus*: bool = True, *autoRenew*: bool = False)

**reject\_wiki\_request** (*requestId*: str)

**reorder\_featured\_users** (*userIds*: list)

**repair\_check\_in** (*method*: int = 0)

**reply\_wall** (*userId*: str, *commentId*: str, *message*: str)

**repost\_blog** (*content*: str = None, *blogId*: str = None, *wikId*: str = None)

**review\_quiz\_questions** (*quizId*: str)

**search\_users** (*nickname*: str, *start*: int = 0, *size*: int = 25)

**send\_action** (*actions*: list, *blogId*: str = None, *quizId*: str = None, *lastAction*: bool = False)

**send\_coins** (*coins*: int, *blogId*: str = None, *chatId*: str = None, *objectId*: str = None, *transactionId*: str = None)

**send\_message** (*chatId*: str, *message*: str = None, *messageType*: int = 0, *file*: BinaryIO = None, *fileType*: str = None, *replyTo*: str = None, *mentionUserIds*: list = None, *stickerId*: str = None, *embedId*: str = None, *embedType*: int = None, *embedLink*: str = None, *embedTitle*: str = None, *embedContent*: str = None, *embedImage*: BinaryIO = None, *linkSnippet*: str = None, *linkSnippetImage*: BinaryIO = None)

Send a Message to a Chat.

#### Parameters

- **message** : Message to be sent

- **chatId** : ID of the Chat.
- **file** : File to be sent.
- **fileType** [Type of the file.]
  - audio, image, gif
- **messageType** : Type of the Message.
- **mentionUserIds** : List of User IDS to mention. '@' needed in the Message.
- **replyTo** : Message ID to reply to.
- **stickerId** : Sticker ID to be sent.
- **embedTitle** : Title of the Embed.
- **embedContent** : Content of the Embed.
- **embedLink** : Link of the Embed.
- **embedImage** : Image of the Embed.
- **embedId** : ID of the Embed.
- **linkSnippet** : Link of the target snippet.
- **linkSnippetImage** : Image target snippet.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**start\_chat** (*userId: Union[str, list], message: str, title: str = None, content: str = None, isGlobal: bool = False, publishToGlobal: bool = False*)  
Start an Chat with an User or List of Users.

#### Parameters

- **userId** : ID of the User or List of User IDs.
- **message** : Starting Message.
- **title** : Title of Group Chat.
- **content** : Content of Group Chat.
- **isGlobal** : If Group Chat is Global.
- **publishToGlobal** : If Group Chat should show in Global.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**strike** (*userId: str, time: int, title: str = None, reason: str = None*)

**submit\_to\_wiki** (*wikiId: str, message: str*)

**subscribe** (*userId: str, autoRenew: str = False, transactionId: str = None*)

**thank\_tip** (*chatId: str, userId: str*)

**transfer\_host** (*chatId: str, userIds: list*)

**transfer\_organizer** (*chatId: str, userIds: list*)

**unban** (*userId: str, reason: str*)

**unblock** (*userId: str*)

Unblock an User.

#### Parameters

- **userId** : ID of the User.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**unfeature** (*userId: str = None, chatId: str = None, blogId: str = None, wikiId: str = None*)

**unfollow** (*userId: str*)

Unfollow an User.

#### Parameters

- **userId** : ID of the User.

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**unhide** (*userId: str = None, chatId: str = None, blogId: str = None, wikiId: str = None, quizId: str = None, fileId: str = None, reason: str = None*)

**unlike\_blog** (*blogId: str = None, wikiId: str = None*)

Remove a like from a Blog or Wiki.

#### Parameters

- **blogId** : ID of the Blog. (for Blogs)
- **wikiId** : ID of the Wiki. (for Wikis)

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**unlike\_comment** (*commentId: str, userId: str = None, blogId: str = None, wikiId: str = None*)

Remove a like from a Comment on a User's Wall, Blog or Wiki.

#### Parameters

- **commentId** : ID of the Comment.
- **userId** : ID of the User. (for Walls)
- **blogId** : ID of the Blog. (for Blogs)
- **wikiId** : ID of the Wiki. (for Wikis)

#### Returns

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**unvote\_comment** (*blogId: str, commentId: str*)

**upvote\_comment** (*blogId: str, commentId: str*)

**vc\_permission** (*chatId: str, permission: int*)

Voice Chat Join Permissions 1 - Open to Everyone 2 - Approval Required 3 - Invite Only

**visit** (*userId: str*)

Visit an User.

**Parameters**

- **userId** : ID of the User.

**Returns**

- **Success** : 200 (int)
- **Fail** : *Exceptions*

**vote\_poll** (*blogId: str, optionId: str*)

**warn** (*userId: str, reason: str = None*)

**class** amino.sub\_client.VCHeaders (*data=None*)

Bases: object

## 2.1.4 Socket

**class** amino.socket.Callbacks (*client*)

Bases: object

**call** (*type, data*)

**default** (*data*)

**event** (*type*)

**on\_avatar\_chat\_end** (*data*)

**on\_avatar\_chat\_not\_answered** (*data*)

**on\_avatar\_chat\_not\_cancelled** (*data*)

**on\_avatar\_chat\_not\_declined** (*data*)

**on\_avatar\_chat\_start** (*data*)

**on\_chat\_background\_changed** (*data*)

**on\_chat\_content\_changed** (*data*)

**on\_chat\_host\_transferred** (*data*)

**on\_chat\_icon\_changed** (*data*)

**on\_chat\_invite** (*data*)

**on\_chat\_pin\_announcement** (*data*)

**on\_chat\_removed\_message** (*data*)

**on\_chat\_tip** (*data*)

**on\_chat\_tipping\_disabled** (*data*)

**on\_chat\_tipping\_enabled** (*data*)

**on\_chat\_title\_changed** (*data*)

**on\_chat\_unpin\_announcement** (*data*)

```

    on_chat_view_only_disabled (data)
    on_chat_view_only_enabled (data)
    on_delete_message (data)
    on_group_member_join (data)
    on_group_member_leave (data)
    on_image_message (data)
    on_invite_message (data)
    on_screen_room_end (data)
    on_screen_room_start (data)
    on_sticker_message (data)
    on_strike_message (data)
    on_text_message (data)
    on_text_message_force_removed (data)
    on_text_message_removed_by_admin (data)
    on_timestamp_message (data)
    on_user_typing_end (data)
    on_user_typing_start (data)
    on_video_chat_end (data)
    on_video_chat_not_answered (data)
    on_video_chat_not_cancelled (data)
    on_video_chat_not_declined (data)
    on_video_chat_start (data)
    on_voice_chat_end (data)
    on_voice_chat_not_answered (data)
    on_voice_chat_not_cancelled (data)
    on_voice_chat_not_declined (data)
    on_voice_chat_permission_invite_only (data)
    on_voice_chat_permission_invited_and_requested (data)
    on_voice_chat_permission_open_to_everyone (data)
    on_voice_chat_start (data)
    on_voice_message (data)
    on_welcome_message (data)
    on_youtube_message (data)
    resolve (data)
class amino.socket.SocketHandler (client, debug=False)
    Bases: object

```

```
close()
handle_message(data)
run()
run_amino_socket()
send(data)
startup()
```

## 2.2 Objects

```
class amino.lib.util.objects.AdminLogList(data)
    Bases: object
    AdminLogList

class amino.lib.util.objects.AvatarFrameList(data)
    Bases: object
    AvatarFrameList

class amino.lib.util.objects.Blog(data)
    Bases: object
    Blog

class amino.lib.util.objects.BlogCategoryList(data)
    Bases: object
    BlogCategoryList

class amino.lib.util.objects.BlogList(data, nextPageToken=None, prevPageToken=None)
    Bases: object
    BlogList

class amino.lib.util.objects.Bubble(data)
    Bases: object
    Bubble

class amino.lib.util.objects.BubbleConfig(data)
    Bases: object
    BubbleConfig

class amino.lib.util.objects.BubbleConfigList(data)
    Bases: object
    BubbleConfigList

class amino.lib.util.objects.BubbleList(data)
    Bases: object
    BubbleList

class amino.lib.util.objects.CommentList(data)
    Bases: object
    CommentList
```



```
class amino.lib.util.objects.Community (data)
    Bases: object

    Community

class amino.lib.util.objects.CommunityList (data)
    Bases: object

    CommunityList

class amino.lib.util.objects.CommunityStats (data)
    Bases: object

    CommunityStats

class amino.lib.util.objects.CommunityStickerCollection (data)
    Bases: object

    CommunityStickerCollection

class amino.lib.util.objects.Event (data)
    Bases: object

    Event

class amino.lib.util.objects.FanClubList (data)
    Bases: object

    FanClubList

class amino.lib.util.objects.FromCode (data)
    Bases: object

    FromCode

class amino.lib.util.objects.GetBlogInfo (data)
    Bases: object

    GetBlogInfo

class amino.lib.util.objects.GetMessages (data)
    Bases: object

    GetMessages

class amino.lib.util.objects.GetSharedFolderInfo (data)
    Bases: object

    GetSharedFolderInfo

class amino.lib.util.objects.GetWikiInfo (data)
    Bases: object

    GetWikiInfo

class amino.lib.util.objects.InfluencerFans (data)
    Bases: object

    InfluencerFans

class amino.lib.util.objects.InviteCode (data)
    Bases: object

    InviteCode

class amino.lib.util.objects.InviteCodeList (data)
    Bases: object
```

**InviteCodeList**

```
class amino.lib.util.objects.JoinRequest(data)
    Bases: object
```

**JoinRequest**

```
class amino.lib.util.objects.LiveLayer(data)
    Bases: object
```

**LiveLayer**

```
class amino.lib.util.objects.LotteryLog(data)
    Bases: object
```

**LotteryLog**

```
class amino.lib.util.objects.Membership(data)
    Bases: object
```

**Membership**

```
class amino.lib.util.objects.Message(data)
    Bases: object
```

**Message**

```
class amino.lib.util.objects.MessageList(data, nextPageToken=None, prevPageToken=None)
    Bases: object
```

**MessageList**

```
class amino.lib.util.objects.NoticeList(data)
    Bases: object
```

**NoticeList**

```
class amino.lib.util.objects.NotificationList(data)
    Bases: object
```

**NotificationList**

```
class amino.lib.util.objects.Objects
    Bases: object
```

**class Users**

```
    Bases: object
```

```
    news_feed = '000000000-0000-0000-0000-0000000000001'
```

```
    team_amino = '000000000-0000-0000-0000-0000000000000'
```

```
class amino.lib.util.objects.QuizAnswers(data)
    Bases: object
```

**QuizAnswers**

```
class amino.lib.util.objects.QuizQuestionList(data)
    Bases: object
```

**QuizQuestionList**

```
class amino.lib.util.objects.QuizRanking(data)
    Bases: object
```

**QuizRanking**

```
class amino.lib.util.objects.QuizRankingList (data)
    Bases: object

    QuizRankingList

class amino.lib.util.objects.QuizRankings (data)
    Bases: object

    QuizRankings

class amino.lib.util.objects.RankingTableList (data)
    Bases: object

    RankingTableList

class amino.lib.util.objects.RecentBlogs (data)
    Bases: object

    RecentBlogs

class amino.lib.util.objects.SharedFolderFile (data)
    Bases: object

    SharedFolderFile

class amino.lib.util.objects.SharedFolderFileList (data)
    Bases: object

    SharedFolderFileList

class amino.lib.util.objects.Sticker (data)
    Bases: object

    Sticker

class amino.lib.util.objects.StickerCollection (data)
    Bases: object

    StickerCollection

class amino.lib.util.objects.StickerCollectionList (data)
    Bases: object

    StickerCollectionList

class amino.lib.util.objects.StickerList (data)
    Bases: object

    StickerList

class amino.lib.util.objects.Thread (data)
    Bases: object

    Thread

class amino.lib.util.objects.ThreadList (data)
    Bases: object

    ThreadList

class amino.lib.util.objects.TippedUsersSummary (data)
    Bases: object

    TippedUsersSummary

class amino.lib.util.objects.UserAchievements (data)
    Bases: object
```

**UserAchievements**

```
class amino.lib.util.objects.UserCheckIns (data)
    Bases: object
```

**UserCheckIns**

```
class amino.lib.util.objects.UserProfile (data)
    Bases: object
```

**UserProfile**

```
class amino.lib.util.objects.UserProfileCountList (data)
    Bases: object
```

**UserProfileCountList**

```
class amino.lib.util.objects.UserProfileList (data)
    Bases: object
```

**UserProfileList**

```
class amino.lib.util.objects.UserSavedBlogs (data)
    Bases: object
```

**UserSavedBlogs**

```
class amino.lib.util.objects.VcReputation (data)
    Bases: object
```

**VcReputation**

```
class amino.lib.util.objects.VisitorsList (data)
    Bases: object
```

**VisitorsList**

```
class amino.lib.util.objects.WalletHistory (data)
    Bases: object
```

**WalletHistory**

```
class amino.lib.util.objects.WalletInfo (data)
    Bases: object
```

**WalletInfo**

```
class amino.lib.util.objects.Wiki (data)
    Bases: object
```

**Wiki**

```
class amino.lib.util.objects.WikiCategory (data)
    Bases: object
```

**WikiCategory**

```
class amino.lib.util.objects.WikiCategoryList (data)
    Bases: object
```

**WikiCategoryList**

```
class amino.lib.util.objects.WikiLabelList (data)
    Bases: object
```

**WikiLabelList**

```

class amino.lib.util.objects.WikiList (data)
    Bases: object

    WikiList

class amino.lib.util.objects.WikiRequestList (data)
    Bases: object

    WikiRequestList

```

## 2.3 Exceptions

```

exception amino.lib.util.exceptions.API_ERR_CHAT_VVCHAT_NO_MORE_REPUTATIONS (**kwargs)
    Bases: Exception

    • API Code : 1627

    • API Message : Unknown Message

    • API String : API_ERR_CHAT_VVCHAT_NO_MORE_REPUTATIONS

exception amino.lib.util.exceptions.API_ERR_COMMUNITY_USER_CREATED_COMMUNITIES_VERIFY (**kwargs)
    Bases: Exception

    • API Code : 257

    • API Message : Unknown Message

    • API String : API_ERR_COMMUNITY_USER_CREATED_COMMUNITIES_VERIFY

exception amino.lib.util.exceptions.API_ERR_EMAIL_NO_PASSWORD (**kwargs)
    Bases: Exception

    • API Code : 251

    • API Message : Unknown Message

    • API String : API_ERR_EMAIL_NO_PASSWORD

exception amino.lib.util.exceptions.API_ERR_INVALID_AUTH_NEW_DEVICE_LINK (**kwargs)
    Bases: Exception

    • API Code : 271

    • API Message : Unknown Message

    • API String : API_ERR_INVALID_AUTH_NEW_DEVICE_LINK

exception amino.lib.util.exceptions.API_ERR_PUSH_SERVER_LIMITATION_APART (**kwargs)
    Bases: Exception

    • API Code : 2501

    • API Message : Unknown Message

    • API String : API_ERR_PUSH_SERVER_LIMITATION_APART

exception amino.lib.util.exceptions.API_ERR_PUSH_SERVER_LIMITATION_COUNT (**kwargs)
    Bases: Exception

    • API Code : 2502

    • API Message : Unknown Message

    • API String : API_ERR_PUSH_SERVER_LIMITATION_COUNT

```

```
exception amino.lib.util.exceptions.API_ERR_PUSH_SERVER_LIMITATION_TIME (**kwargs)
    Bases: Exception
        • API Code : 2504
        • API Message : Unknown Message
        • API String : API_ERR_PUSH_SERVER_LIMITATION_TIME

exception amino.lib.util.exceptions.API_ERR_PUSH_SERVER_LINK_NOT_IN_COMMUNITY (**kwargs)
    Bases: Exception
        • API Code : 2503
        • API Message : Unknown Message
        • API String : API_ERR_PUSH_SERVER_LINK_NOT_IN_COMMUNITY

exception amino.lib.util.exceptions.AccessDenied (**kwargs)
    Bases: Exception
        • API Code : 106
        • API Message : Access denied.
        • API String : Unknown String

exception amino.lib.util.exceptions.AccountAlreadyRestored (**kwargs)
    Bases: Exception
        • API Code : 2800
        • API Message : Account already restored.
        • API String : Unknown String

exception amino.lib.util.exceptions.AccountDeleted (**kwargs)
    Bases: Exception
        • API Code : 246
        • API Message : Unknown Message
        • API String : AUTH_RECOVERABLE_DELETED_ACCOUNT

exception amino.lib.util.exceptions.AccountDisabled (**kwargs)
    Bases: Exception
        • API Code : 210
        • API Message : This account is disabled.
        • API String : AUTH_DISABLED_ACCOUNT

exception amino.lib.util.exceptions.AccountDoesntExist (**kwargs)
    Bases: Exception
        • API Code : 216
        • API Message : Unknown Message
        • API String : AUTH_ACCOUNT_NOT_EXISTS

exception amino.lib.util.exceptions.AccountLimitReached (**kwargs)
    Bases: Exception
        • API Code : 219
```

- **API Message** : A maximum of 3 accounts can be created from this device. If you forget your password, please reset it.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**ActionNotAllowed** (\*\*kwargs)  
Bases: Exception

- **API Code** : 110
- **API Message** : Action not allowed.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**ActivateAccount** (\*\*kwargs)  
Bases: Exception

- **API Code** : 238
- **API Message** : Please activate your account first. Check your email, including your spam folder.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**AgeTooLow** (\*\*kwargs)  
Bases: Exception

Raised when you try to configure an account but the age is too low. Minimum is 13.

**exception** amino.lib.util.exceptions.**AlreadyCheckedIn** (\*\*kwargs)  
Bases: Exception

- **API Code** : 2601
- **API Message** : Sorry, you can't check in any more.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**AlreadyPlayedLottery** (\*\*kwargs)  
Bases: Exception

- **API Code** : 4400
- **API Message** : You have played the maximum number of lucky draws.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**AlreadyRequestedJoinCommunity** (\*\*kwargs)  
Bases: Exception

- **API Code** : 2001
- **API Message** : Sorry, you have already submitted a membership request.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**AlreadyUsedMonthlyRepair** (\*\*kwargs)  
Bases: Exception

- **API Code** : 2611
- **API Message** : Unknown Message
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**AminoIDAlreadyChanged** (\*\*kwargs)  
Bases: Exception

- **API Code** : 6001

- **API Message** : Amino ID cannot be changed after you set it.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**BadImage** (\*\*kwargs)

Bases: Exception

- **API Code** : 300
- **API Message** : Unknown Message
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**CannotFetchImage** (\*\*kwargs)

Bases: Exception

Raised when an image cannot be fetched.

**exception** amino.lib.util.exceptions.**CannotSendCoins** (\*\*kwargs)

Bases: Exception

- **API Code** : 4500, 4501
- **API Message** : Unknown Message
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**CantFollowYourself** (\*\*kwargs)

Bases: Exception

- **API Code** : 221
- **API Message** : Unknown Message
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**CantLeaveCommunity** (\*\*kwargs)

Bases: Exception

- **API Code** : 239
- **API Message** : Sorry, you can not do this before transferring your Agent status to another member.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**ChatFull** (\*\*kwargs)

Bases: Exception

- **API Code** : 1605
- **API Message** : Unknown Message
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**ChatInvitesDisabled** (\*\*kwargs)

Bases: Exception

- **API Code** : 1611
- **API Message** : This user has disabled chat invite requests.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**ChatMessageTooBig** (\*\*kwargs)

Bases: Exception

- **API Code** : 1664
- **API Message** : Unknown Message



- **API String** : API\_ERR\_CHAT\_MESSAGE\_CONTENT\_TOO\_LONG

**exception** amino.lib.util.exceptions.**ChatRequestsBlocked** (\*\*kwargs)

Bases: Exception

Raised when you try to make an action but the end user has chat requests blocked.

**exception** amino.lib.util.exceptions.**ChatViewOnly** (\*\*kwargs)

Bases: Exception

- **API Code** : 1663
- **API Message** : Unknown Message
- **API String** : Unknown String

amino.lib.util.exceptions.**CheckException** (data)

**exception** amino.lib.util.exceptions.**CommandCooldown** (\*\*kwargs)

Bases: Exception

- **API Code** : 291
- **API Message** : Whoa there! You've done too much too quickly. Take a break and try again later.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**CommunityCreateLimitReached** (\*\*kwargs)

Bases: Exception

- **API Code** : 806
- **API Message** : Unknown Message
- **API String** : API\_ERR\_COMMUNITY\_USER\_CREATED\_COMMUNITIES\_EXCEED\_QUOTA

**exception** amino.lib.util.exceptions.**CommunityDeleted** (\*\*kwargs)

Bases: Exception

- **API Code** : 833
- **API Message** : This Community has been deleted.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**CommunityDisabled** (\*\*kwargs)

Bases: Exception

- **API Code** : 814
- **API Message** : This Community is disabled.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**CommunityNameAlreadyTaken** (\*\*kwargs)

Bases: Exception

- **API Code** : 805
- **API Message** : Unknown Message
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**CommunityNeeded** (\*\*kwargs)

Bases: Exception

Raised when you try to execute an command but a Community needs to be specified.

**exception** amino.lib.util.exceptions.**CommunityNoLongerExists** (\*\*kwargs)

Bases: Exception

- **API Code** : 801
- **API Message** : This Community no longer exists.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**CommunityNotFound** (\*\*kwargs)

Bases: Exception

Raised when you search for a community but nothing is found.

**exception** amino.lib.util.exceptions.**DuplicatePollOption** (\*\*kwargs)

Bases: Exception

- **API Code** : 1501
- **API Message** : Sorry, you have duplicate poll options.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**EmailAlreadyTaken** (\*\*kwargs)

Bases: Exception

- **API Code** : 215
- **API Message** : Hey this email X has been registered already. You can try to log in with the email or edit the email.
- **API String** : API\_ERR\_EMAIL\_TAKEN

**exception** amino.lib.util.exceptions.**EmailFlaggedAsSpam** (\*\*kwargs)

Bases: Exception

- **API Code** : 241
- **API Message** : This email provider has been flagged for use in spamming.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**FailedLogin** (\*\*kwargs)

Bases: Exception

Raised when you try to login but it fails.

**exception** amino.lib.util.exceptions.**FileTooLarge** (\*\*kwargs)

Bases: Exception

- **API Code** : 102
- **API Message** : Unknown Message
- **API String** : API\_STD\_ERR\_ENTITY\_TOO\_LARGE\_RAW

**exception** amino.lib.util.exceptions.**FlagTypeNeeded** (\*\*kwargs)

Bases: Exception

Raised when you try to flag a community, blog or user but a Flag Type needs to be specified.

**exception** amino.lib.util.exceptions.**IncorrectVerificationCode** (\*\*kwargs)

Bases: Exception

- **API Code** : 3102
- **API Message** : Incorrect verification code.

- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**InsufficientLevel** (\*\*kwargs)  
Bases: Exception

- **API Code**: 551
- **API Message**: This post type is restricted to members with a level X ranking and above.
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**InvalidAccountOrPassword** (\*\*kwargs)  
Bases: Exception

- **API Code**: 200
- **API Message**: Unknown Message
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**InvalidAminoID** (\*\*kwargs)  
Bases: Exception

- **API Code**: 6002
- **API Message**: Invalid Amino ID
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**InvalidCodeOrLink** (\*\*kwargs)  
Bases: Exception

- **API Code**: 802
- **API Message**: Sorry, this code or link is invalid.
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**InvalidDevice** (\*\*kwargs)  
Bases: Exception

- **API Code**: 218
- **API Message**: Error! Your device is currently not supported, or the app is out of date. Please update to the latest version.
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**InvalidEmail** (\*\*kwargs)  
Bases: Exception

- **API Code**: 213
- **API Message**: Invalid email address.
- **API String**: API\_ERR\_EMAIL

**exception** amino.lib.util.exceptions.**InvalidName** (\*\*kwargs)  
Bases: Exception

- **API Code**: 99001
- **API Message**: Sorry, the name is invalid.
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**InvalidPassword** (\*\*kwargs)  
Bases: Exception

- **API Code** : 214
- **API Message** : Invalid password. Password must be 6 characters or more and contain no spaces.
- **API String** : API\_ERR\_PASSWORD

**exception** amino.lib.util.exceptions.**InvalidRequest** (\*\*kwargs)

Bases: Exception

- **API Code** : 103, 104
- **API Message** : Invalid Request. Please update to the latest version. If the problem continues, please contact us.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**InvalidSession** (\*\*kwargs)

Bases: Exception

- **API Code** : 105
- **API Message** : Unknown Message
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**InvalidThemepack** (\*\*kwargs)

Bases: Exception

- **API Code** : 313
- **API Message** : Unknown Message
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**InvalidVoiceNote** (\*\*kwargs)

Bases: Exception

- **API Code** : 314
- **API Message** : Unknown Message
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**InviteCodeNotFound** (\*\*kwargs)

Bases: Exception

- **API Code** : 1900
- **API Message** : Sorry, the requested data no longer exists. Try refreshing the view.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**LevelFiveRequiredToEnableProps** (\*\*kwargs)

Bases: Exception

- **API Code** : 1661
- **API Message** : Unknown Message
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**LibraryUpdateAvailable** (\*\*kwargs)

Bases: Exception

Raised when a new library update is available.

**exception** amino.lib.util.exceptions.**MemberKickedByOrganizer** (\*\*kwargs)

Bases: Exception

- **API Code** : 1637
- **API Message** : This member was previously kicked by the organizer and cannot be reinvited.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**MessageNeeded** (\*\*kwargs)

Bases: Exception

- **API Code** : 113
- **API Message** : Be more specific, please.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**NoChatThread** (\*\*kwargs)

Bases: Exception

Raised when you try to make an action but no chat was selected.

**exception** amino.lib.util.exceptions.**NoCommunity** (\*\*kwargs)

Bases: Exception

Raised when you try to make an action but no community was selected.

**exception** amino.lib.util.exceptions.**NoImageSource** (\*\*kwargs)

Bases: Exception

Raised when you try to make an action but no image source was selected.

**exception** amino.lib.util.exceptions.**NotEnoughCoins** (\*\*kwargs)

Bases: Exception

- **API Code** : 4300
- **API Message** : Unknown Message
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**NotLoggedIn** (\*\*kwargs)

Bases: Exception

Raised when you try to make an action but you aren't logged in.

**exception** amino.lib.util.exceptions.**NotOwnerOfChatBubble** (\*\*kwargs)

Bases: Exception

- **API Code** : 3905
- **API Message** : You are not the owner of this chat bubble.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**PageReportedTooRecently** (\*\*kwargs)

Bases: Exception

- **API Code** : 503
- **API Message** : Sorry, you have reported this page too recently.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**ReachedMaxCategories** (\*\*kwargs)

Bases: Exception

- **API Code** : 1002
- **API Message** : Sorry, you can create up to 100 categories.

- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**ReachedMaxPollOptions** (\*\*kwargs)

Bases: Exception

- **API Code**: 1507
- **API Message**: Sorry, you can only join or add up to 5 of your items per poll.
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**ReachedMaxTitles** (\*\*kwargs)

Bases: Exception

- **API Code**: 262
- **API Message**: You can only add up to 20 Titles. Please choose the most relevant ones.
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**ReachedTitleLength** (\*\*kwargs)

Bases: Exception

- **API Code**: 240
- **API Message**: Sorry, the max length of member's title is limited to 20.
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**ReasonNeeded** (\*\*kwargs)

Bases: Exception

Raised when you try to execute an command but a Reason needs to be specified.

**exception** amino.lib.util.exceptions.**RemovedFromChat** (\*\*kwargs)

Bases: Exception

- **API Code**: 1612
- **API Message**: You've been removed from this chatroom.
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**RequestRejected** (\*\*kwargs)

Bases: Exception

- **API Code**: 235
- **API Message**: Request rejected. You have been temporarily muted (read only mode) because you have received a strike. To learn more, please check the Help Center.
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**RequestedNoLongerExists** (\*\*kwargs)

Bases: Exception

- **API Code**: 500, 700, 1600
- **API Message**: Sorry, the requested data no longer exists. Try refreshing the view.
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**ServiceUnderMaintenance** (\*\*kwargs)

Bases: Exception

- **API Code**: 111
- **API Message**: Sorry, this service is under maintenance. Please check back later.

- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**SpecifyType** (\*\*kwargs)  
Bases: Exception

Raised when you need to specify the output of the command.

**exception** amino.lib.util.exceptions.**TooManyChats** (\*\*kwargs)  
Bases: Exception

- **API Code**: 1602
- **API Message**: Sorry, you can only have up to 1000 chat sessions.
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**TooManyInviteUsers** (\*\*kwargs)  
Bases: Exception

- **API Code**: 1606
- **API Message**: Sorry, you can only invite up to 999 people.
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**TooManyRequests** (\*\*kwargs)  
Bases: Exception

- **API Code**: 219
- **API Message**: Too many requests. Try again later.
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**TransferRequestNeeded** (\*\*kwargs)  
Bases: Exception

Raised when you need to transfer host to complete the action.

**exception** amino.lib.util.exceptions.**UnexistentData** (\*\*kwargs)  
Bases: Exception

- **API Code**: 107
- **API Message**: The requested data does not exist.
- **API String**: Unknown String

**exception** amino.lib.util.exceptions.**UnknownResponse** (\*\*kwargs)  
Bases: Exception

Raised when an error occurs but the reason is unknown.

**exception** amino.lib.util.exceptions.**UnsupportedEmail** (\*\*kwargs)  
Bases: Exception

- **API Code**: 215
- **API Message**: This email address is not supported.
- **API String**: API\_ERR\_EMAIL\_TAKEN

**exception** amino.lib.util.exceptions.**UnsupportedLanguage** (\*\*kwargs)  
Bases: Exception

Raised when you try to use a language that isn't supported or exists.

**exception** amino.lib.util.exceptions.**UnsupportedService** (\*\*kwargs)

Bases: Exception

- **API Code** : 100
- **API Message** : Unsupported service. Your client may be out of date. Please update it to the latest version.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**UserBannedByTeamAmino** (\*\*kwargs)

Bases: Exception

- **API Code** : 293
- **API Message** : Sorry, this user has been banned by Team Amino.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**UserNotJoined** (\*\*kwargs)

Bases: Exception

- **API Code** : 1613
- **API Message** : Sorry, this user has not joined.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**UserNotMemberOfCommunity** (\*\*kwargs)

Bases: Exception

- **API Code** : 230
- **API Message** : You have to join this Community first.
- **API String** : API\_ERR\_USER\_NOT\_IN\_COMMUNITY

**exception** amino.lib.util.exceptions.**UserUnavailable** (\*\*kwargs)

Bases: Exception

- **API Code** : 225
- **API Message** : This user is unavailable.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**VerificationRequired** (\*\*kwargs)

Bases: Exception

- **API Code** : 270
- **API Message** : Verification Required.
- **API String** : API\_ERR\_NEED\_TWO\_FACTOR\_AUTHENTICATION

**exception** amino.lib.util.exceptions.**WallCommentingDisabled** (\*\*kwargs)

Bases: Exception

- **API Code** : 702
- **API Message** : This member has disabled commenting on their wall.
- **API String** : Unknown String

**exception** amino.lib.util.exceptions.**WrongType** (\*\*kwargs)

Bases: Exception

Raised when you attribute the function the wrong type.



```
exception amino.lib.util.exceptions.YouAreBanned (**kwargs)
    Bases: Exception
    • API Code : 229
    • API Message : You are banned.
    • API String : Unknown String
```

## 2.4 Device

```
class amino.lib.util.device.DeviceGenerator (deviceId=None)
    Bases: object
```

## 2.5 Headers

```
class amino.lib.util.headers.Headers (data=None, type=None, deviceId: str = None, sig: str
                                         = None)
    Bases: object
```

## 2.6 Helpers

```
amino.lib.util.helpers.decode_sid (sid: str) → dict
amino.lib.util.helpers.generate_device_info ()
amino.lib.util.helpers.sid_to_ip_address (SID: str) → str
amino.lib.util.helpers.sid_to_uid (SID: str) → str
```



## CHAPTER 3

---

### Contact

---

Here you can contact me about the library.

- Developer : Slimakoi#6422 (<https://linktr.ee/Slimakoi>)
- Discord Support Server : <https://rebrand.ly/slimakoi-and-friends>



## CHAPTER 4

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### a

- `amino.acm`, 3
- `amino.client`, 4
- `amino.lib.util.device`, 53
- `amino.lib.util.exceptions`, 41
- `amino.lib.util.headers`, 53
- `amino.lib.util.helpers`, 53
- `amino.lib.util.objects`, 36
- `amino.socket`, 34
- `amino.sub_client`, 21





## A

accept\_host() (*amino.client.Client method*), 4  
 accept\_host() (*amino.sub\_client.SubClient method*), 21  
 accept\_join\_request() (*amino.acm.ACM method*), 3  
 accept\_organizer() (*amino.client.Client method*), 4  
 accept\_organizer() (*amino.sub\_client.SubClient method*), 21  
 accept\_wiki\_request() (*amino.sub\_client.SubClient method*), 21  
 AccessDenied, 42  
 AccountAlreadyRestored, 42  
 AccountDeleted, 42  
 AccountDisabled, 42  
 AccountDoesntExist, 42  
 AccountLimitReached, 43  
 ACM (*class in amino.acm*), 3  
 ActionNotAllowed, 43  
 activate\_account() (*amino.client.Client method*), 4  
 activate\_bubble() (*amino.sub\_client.SubClient method*), 21  
 ActivateAccount, 43  
 activity\_status() (*amino.sub\_client.SubClient method*), 21  
 add\_influencer() (*amino.acm.ACM method*), 3  
 add\_linked\_community() (*amino.client.Client method*), 4  
 add\_poll\_option() (*amino.sub\_client.SubClient method*), 21  
 add\_to\_favorites() (*amino.sub\_client.SubClient method*), 21  
 AdminLogList (*amino.lib.util.objects.AdminLogList attribute*), 36  
 AdminLogList (*class in amino.lib.util.objects*), 36  
 AgeTooLow, 43  
 AlreadyCheckedIn, 43  
 AlreadyPlayedLottery, 43  
 AlreadyRequestedJoinCommunity, 43  
 AlreadyUsedMonthlyRepair, 43  
 amino.acm (*module*), 3  
 amino.client (*module*), 4  
 amino.lib.util.device (*module*), 53  
 amino.lib.util.exceptions (*module*), 41  
 amino.lib.util.headers (*module*), 53  
 amino.lib.util.helpers (*module*), 53  
 amino.lib.util.objects (*module*), 36  
 amino.socket (*module*), 34  
 amino.sub\_client (*module*), 21  
 AminoIDAlreadyChanged, 43  
 API\_ERR\_CHAT\_VVCHAT\_NO\_MORE\_REPUTATIONS, 41  
 API\_ERR\_COMMUNITY\_USER\_CREATED\_COMMUNITIES\_VERIFY, 41  
 API\_ERR\_EMAIL\_NO\_PASSWORD, 41  
 API\_ERR\_INVALID\_AUTH\_NEW\_DEVICE\_LINK, 41  
 API\_ERR\_PUSH\_SERVER\_LIMITATION\_APART, 41  
 API\_ERR\_PUSH\_SERVER\_LIMITATION\_COUNT, 41  
 API\_ERR\_PUSH\_SERVER\_LIMITATION\_TIME, 41  
 API\_ERR\_PUSH\_SERVER\_LINK\_NOT\_IN\_COMMUNITY, 42  
 apply\_avatar\_frame() (*amino.sub\_client.SubClient method*), 21  
 apply\_bubble() (*amino.sub\_client.SubClient method*), 22  
 AvatarFrameList (*amino.lib.util.objects.AvatarFrameList attribute*), 36  
 AvatarFrameList (*class in amino.lib.util.objects*), 36

## B

BadImage, 44  
 ban() (*amino.sub\_client.SubClient method*), 22  
 block() (*amino.client.Client method*), 4  
 block() (*amino.sub\_client.SubClient method*), 22

Blog (*amino.lib.util.objects.Blog* attribute), 36  
Blog (*class in amino.lib.util.objects*), 36  
BlogCategoryList (*amino.lib.util.objects.BlogCategoryList* attribute), 36  
BlogCategoryList (*class in amino.lib.util.objects*), 36  
BlogList (*amino.lib.util.objects.BlogList* attribute), 36  
BlogList (*class in amino.lib.util.objects*), 36  
Bubble (*amino.lib.util.objects.Bubble* attribute), 36  
Bubble (*class in amino.lib.util.objects*), 36  
BubbleConfig (*amino.lib.util.objects.BubbleConfig* attribute), 36  
BubbleConfig (*class in amino.lib.util.objects*), 36  
BubbleConfigList (*amino.lib.util.objects.BubbleConfigList* attribute), 36  
BubbleConfigList (*class in amino.lib.util.objects*), 36  
BubbleList (*amino.lib.util.objects.BubbleList* attribute), 36  
BubbleList (*class in amino.lib.util.objects*), 36

## C

call () (*amino.socket.Callbacks* method), 34  
Callbacks (*class in amino.socket*), 34  
CannotFetchImage, 44  
CannotSendCoins, 44  
CantFollowYourself, 44  
CantLeaveCommunity, 44  
change\_guidelines () (*amino.acm.ACM* method), 3  
change\_module () (*amino.acm.ACM* method), 3  
change\_password () (*amino.client.Client* method), 4  
change\_sidepanel\_color () (*amino.acm.ACM* method), 3  
change\_welcome\_message () (*amino.acm.ACM* method), 3  
ChatFull, 44  
ChatInvitesDisabled, 44  
ChatMessageTooBig, 44  
ChatRequestsBlocked, 45  
ChatViewOnly, 45  
check\_device () (*amino.client.Client* method), 5  
check\_in () (*amino.sub\_client.SubClient* method), 22  
check\_notifications () (*amino.sub\_client.SubClient* method), 22  
CheckException () (*in module amino.lib.util.exceptions*), 45  
claim\_new\_user\_coupon () (*amino.client.Client* method), 5  
claim\_vc\_reputation () (*amino.sub\_client.SubClient* method), 22  
clear\_notifications () (*amino.sub\_client.SubClient* method), 22  
Client (*class in amino.client*), 4

close () (*amino.socket.SocketHandler* method), 35  
CommandCooldown, 45  
comment () (*amino.client.Client* method), 5  
comment () (*amino.sub\_client.SubClient* method), 22  
CommentList (*amino.lib.util.objects.CommentList* attribute), 36  
CommentList (*class in amino.lib.util.objects*), 36  
Community (*amino.lib.util.objects.Community* attribute), 37  
Community (*class in amino.lib.util.objects*), 36  
CommunityCreateLimitReached, 45  
CommunityDeleted, 45  
CommunityDisabled, 45  
CommunityList (*amino.lib.util.objects.CommunityList* attribute), 37  
CommunityList (*class in amino.lib.util.objects*), 37  
CommunityNameAlreadyTaken, 45  
CommunityNeeded, 45  
CommunityNoLongerExists, 45  
CommunityNotFound, 46  
CommunityStats (*amino.lib.util.objects.CommunityStats* attribute), 37  
CommunityStats (*class in amino.lib.util.objects*), 37  
CommunityStickerCollection (*amino.lib.util.objects.CommunityStickerCollection* attribute), 37  
CommunityStickerCollection (*class in amino.lib.util.objects*), 37  
configure () (*amino.client.Client* method), 5  
create\_community () (*amino.acm.ACM* method), 3  
create\_shared\_folder () (*amino.sub\_client.SubClient* method), 22  
create\_wiki\_category () (*amino.sub\_client.SubClient* method), 22

## D

deactivate\_bubble () (*amino.sub\_client.SubClient* method), 22  
decode\_sid () (*in module amino.lib.util.helpers*), 53  
default () (*amino.socket.Callbacks* method), 34  
delete\_account () (*amino.client.Client* method), 6  
delete\_blog () (*amino.sub\_client.SubClient* method), 22  
delete\_chat () (*amino.sub\_client.SubClient* method), 22  
delete\_comment () (*amino.client.Client* method), 6  
delete\_comment () (*amino.sub\_client.SubClient* method), 23  
delete\_community () (*amino.acm.ACM* method), 3  
delete\_invite\_code () (*amino.sub\_client.SubClient* method), 23  
delete\_message () (*amino.client.Client* method), 6  
delete\_message () (*amino.sub\_client.SubClient* method), 23

delete\_notification() *(amino.sub\_client.SubClient method)*, 23  
 delete\_pending\_role() *(amino.acm.ACM method)*, 3  
 delete\_wiki() *(amino.sub\_client.SubClient method)*, 23  
 DeviceGenerator *(class in amino.lib.util.device)*, 53  
 downvote\_comment() *(amino.sub\_client.SubClient method)*, 23  
 DuplicatePollOption, 46

## E

edit\_blog() *(amino.sub\_client.SubClient method)*, 23  
 edit\_chat() *(amino.client.Client method)*, 6  
 edit\_chat() *(amino.sub\_client.SubClient method)*, 23  
 edit\_community() *(amino.acm.ACM method)*, 3  
 edit\_profile() *(amino.client.Client method)*, 7  
 edit\_profile() *(amino.sub\_client.SubClient method)*, 24  
 edit\_titles() *(amino.sub\_client.SubClient method)*, 24  
 EmailAlreadyTaken, 46  
 EmailFlaggedAsSpam, 46  
 end\_vc() *(amino.client.Client method)*, 7  
 Event *(amino.lib.util.objects.Event attribute)*, 37  
 Event *(class in amino.lib.util.objects)*, 37  
 event() *(amino.socket.Callbacks method)*, 34

## F

FailedLogin, 46  
 FanClubList *(amino.lib.util.objects.FanClubList attribute)*, 37  
 FanClubList *(class in amino.lib.util.objects)*, 37  
 feature() *(amino.sub\_client.SubClient method)*, 24  
 FileTooLarge, 46  
 flag() *(amino.client.Client method)*, 7  
 flag() *(amino.sub\_client.SubClient method)*, 24  
 flag\_community() *(amino.client.Client method)*, 8  
 FlagTypeNeeded, 46  
 follow() *(amino.client.Client method)*, 8  
 follow() *(amino.sub\_client.SubClient method)*, 25  
 FromCode *(amino.lib.util.objects.FromCode attribute)*, 37  
 FromCode *(class in amino.lib.util.objects)*, 37

## G

generate\_device\_info() *(in module amino.lib.util.helpers)*, 53  
 generate\_invite\_code() *(amino.sub\_client.SubClient method)*, 25  
 get\_account\_info() *(amino.client.Client method)*, 8

get\_all\_users() *(amino.client.Client method)*, 8  
 get\_all\_users() *(amino.sub\_client.SubClient method)*, 25  
 get\_avatar\_frames() *(amino.client.Client method)*, 8  
 get\_best\_quiz() *(amino.sub\_client.SubClient method)*, 25  
 get\_blocked\_users() *(amino.client.Client method)*, 8  
 get\_blocked\_users() *(amino.sub\_client.SubClient method)*, 25  
 get\_blocker\_users() *(amino.client.Client method)*, 9  
 get\_blocker\_users() *(amino.sub\_client.SubClient method)*, 25  
 get\_blog\_categories() *(amino.sub\_client.SubClient method)*, 26  
 get\_blog\_comments() *(amino.client.Client method)*, 9  
 get\_blog\_comments() *(amino.sub\_client.SubClient method)*, 26  
 get\_blog\_info() *(amino.client.Client method)*, 9  
 get\_blog\_info() *(amino.sub\_client.SubClient method)*, 26  
 get\_blog\_users() *(amino.sub\_client.SubClient method)*, 26  
 get\_blogs\_by\_category() *(amino.sub\_client.SubClient method)*, 26  
 get\_bubble\_info() *(amino.sub\_client.SubClient method)*, 26  
 get\_bubble\_template\_list() *(amino.sub\_client.SubClient method)*, 26  
 get\_categories() *(amino.acm.ACM method)*, 3  
 get\_chat\_messages() *(amino.client.Client method)*, 9  
 get\_chat\_messages() *(amino.sub\_client.SubClient method)*, 26  
 get\_chat\_thread() *(amino.client.Client method)*, 9  
 get\_chat\_thread() *(amino.sub\_client.SubClient method)*, 26  
 get\_chat\_threads() *(amino.client.Client method)*, 9  
 get\_chat\_threads() *(amino.sub\_client.SubClient method)*, 26  
 get\_chat\_users() *(amino.client.Client method)*, 9  
 get\_chat\_users() *(amino.sub\_client.SubClient method)*, 26  
 get\_community\_info() *(amino.client.Client method)*, 10  
 get\_community\_stats() *(amino.acm.ACM method)*, 3  
 get\_community\_stickers() *(amino.sub\_client.SubClient method)*, 27  
 get\_community\_user\_stats() *(amino.acm.ACM method)*, 27

*method*), 3  
get\_eventlog() (*amino.client.Client method*), 10  
get\_featured\_users() (*amino.sub\_client.SubClient method*), 27  
get\_from\_code() (*amino.client.Client method*), 10  
get\_from\_deviceid() (*amino.client.Client method*), 10  
get\_from\_id() (*amino.client.Client method*), 10  
get\_hidden\_blogs() (*amino.sub\_client.SubClient method*), 27  
get\_influencer\_fans() (*amino.sub\_client.SubClient method*), 27  
get\_invite\_codes() (*amino.sub\_client.SubClient method*), 27  
get\_join\_requests() (*amino.acm.ACM method*), 3  
get\_leaderboard\_info() (*amino.sub\_client.SubClient method*), 27  
get\_linked\_communities() (*amino.client.Client method*), 10  
get\_live\_layer() (*amino.sub\_client.SubClient method*), 27  
get\_membership\_info() (*amino.client.Client method*), 11  
get\_message\_info() (*amino.client.Client method*), 11  
get\_message\_info() (*amino.sub\_client.SubClient method*), 27  
get\_notice\_list() (*amino.acm.ACM method*), 3  
get\_notices() (*amino.sub\_client.SubClient method*), 27  
get\_notifications() (*amino.sub\_client.SubClient method*), 27  
get\_online\_favorite\_users() (*amino.sub\_client.SubClient method*), 27  
get\_online\_users() (*amino.sub\_client.SubClient method*), 27  
get\_public\_chat\_threads() (*amino.sub\_client.SubClient method*), 27  
get\_quiz\_rankings() (*amino.sub\_client.SubClient method*), 27  
get\_recent\_blogs() (*amino.sub\_client.SubClient method*), 27  
get\_recent\_quiz() (*amino.sub\_client.SubClient method*), 27  
get\_recent\_wiki\_items() (*amino.sub\_client.SubClient method*), 27  
get\_saved\_blogs() (*amino.sub\_client.SubClient method*), 27  
get\_shared\_folder\_files() (*amino.sub\_client.SubClient method*), 27  
get\_shared\_folder\_info() (*amino.sub\_client.SubClient method*), 27  
get\_sticker\_collection() (*amino.sub\_client.SubClient method*), 28  
get\_sticker\_pack\_info() (*amino.sub\_client.SubClient method*), 28  
get\_sticker\_packs() (*amino.sub\_client.SubClient method*), 28  
get\_store\_chat\_bubbles() (*amino.sub\_client.SubClient method*), 28  
get\_store\_stickers() (*amino.sub\_client.SubClient method*), 28  
get\_subscriptions() (*amino.client.Client method*), 11  
get\_supported\_languages() (*amino.client.Client method*), 11  
get\_ta\_announcements() (*amino.client.Client method*), 11  
get\_tipped\_users() (*amino.sub\_client.SubClient method*), 28  
get\_trending\_quiz() (*amino.sub\_client.SubClient method*), 28  
get\_unlinked\_communities() (*amino.client.Client method*), 12  
get\_user\_achievements() (*amino.sub\_client.SubClient method*), 28  
get\_user\_blogs() (*amino.sub\_client.SubClient method*), 28  
get\_user\_checkins() (*amino.sub\_client.SubClient method*), 28  
get\_user\_followers() (*amino.client.Client method*), 12  
get\_user\_followers() (*amino.sub\_client.SubClient method*), 28  
get\_user\_following() (*amino.client.Client method*), 12  
get\_user\_following() (*amino.sub\_client.SubClient method*), 28  
get\_user\_info() (*amino.client.Client method*), 12  
get\_user\_info() (*amino.sub\_client.SubClient method*), 28  
get\_user\_visitors() (*amino.client.Client method*), 12  
get\_user\_visitors() (*amino.sub\_client.SubClient method*), 28  
get\_user\_wikis() (*amino.sub\_client.SubClient method*), 29  
get\_vc\_reputation\_info() (*amino.sub\_client.SubClient method*), 29  
get\_wall\_comments() (*amino.client.Client method*), 13  
get\_wall\_comments() (*amino.sub\_client.SubClient method*), 29  
get\_wallet\_history() (*amino.client.Client method*), 13  
get\_wallet\_info() (*amino.client.Client method*), 13

[get\\_wiki\\_categories\(\)](#) ([amino.sub\\_client.SubClient method](#)), 29  
[get\\_wiki\\_category\(\)](#) ([amino.sub\\_client.SubClient method](#)), 29  
[get\\_wiki\\_info\(\)](#) ([amino.sub\\_client.SubClient method](#)), 29  
[get\\_wiki\\_submissions\(\)](#) ([amino.sub\\_client.SubClient method](#)), 29  
[GetBlogInfo](#) ([amino.lib.util.objects.GetBlogInfo attribute](#)), 37  
[GetBlogInfo](#) ([class in amino.lib.util.objects](#)), 37  
[GetMessages](#) ([amino.lib.util.objects.GetMessages attribute](#)), 37  
[GetMessages](#) ([class in amino.lib.util.objects](#)), 37  
[GetSharedFolderInfo](#) ([amino.lib.util.objects.GetSharedFolderInfo attribute](#)), 37  
[GetSharedFolderInfo](#) ([class in amino.lib.util.objects](#)), 37  
[GetWikiInfo](#) ([amino.lib.util.objects.GetWikiInfo attribute](#)), 37  
[GetWikiInfo](#) ([class in amino.lib.util.objects](#)), 37

## H

[handle\\_message\(\)](#) ([amino.socket.SocketHandler method](#)), 36  
[handle\\_socket\\_message\(\)](#) ([amino.client.Client method](#)), 13  
[Headers](#) ([class in amino.lib.util.headers](#)), 53  
[hide\(\)](#) ([amino.sub\\_client.SubClient method](#)), 29

## I

[IncorrectVerificationCode](#), 46  
[InfluencerFans](#) ([amino.lib.util.objects.InfluencerFans attribute](#)), 37  
[InfluencerFans](#) ([class in amino.lib.util.objects](#)), 37  
[InsufficientLevel](#), 47  
[InvalidAccountOrPassword](#), 47  
[InvalidAminoID](#), 47  
[InvalidCodeOrLink](#), 47  
[InvalidDevice](#), 47  
[InvalidEmail](#), 47  
[InvalidName](#), 47  
[InvalidPassword](#), 27  
[InvalidRequest](#), 48  
[InvalidSession](#), 48  
[InvalidThemepack](#), 48  
[InvalidVoiceNote](#), 48  
[invite\\_to\\_chat\(\)](#) ([amino.client.Client method](#)), 13  
[invite\\_to\\_chat\(\)](#) ([amino.sub\\_client.SubClient method](#)), 29  
[invite\\_to\\_vc\(\)](#) ([amino.client.Client method](#)), 14  
[invite\\_to\\_vc\(\)](#) ([amino.sub\\_client.SubClient method](#)), 29

[InviteCode](#) ([amino.lib.util.objects.InviteCode attribute](#)), 37  
[InviteCode](#) ([class in amino.lib.util.objects](#)), 37  
[InviteCodeList](#) ([amino.lib.util.objects.InviteCodeList attribute](#)), 37  
[InviteCodeList](#) ([class in amino.lib.util.objects](#)), 37  
[InviteCodeNotFound](#), 48

## J

[join\\_chat\(\)](#) ([amino.client.Client method](#)), 14  
[join\\_chat\(\)](#) ([amino.sub\\_client.SubClient method](#)), 30  
[join\\_community\(\)](#) ([amino.client.Client method](#)), 14  
[join\\_video\\_chat\(\)](#) ([amino.client.Client method](#)), 14  
[join\\_video\\_chat\\_as\\_viewer\(\)](#) ([amino.client.Client method](#)), 14  
[join\\_voice\\_chat\(\)](#) ([amino.client.Client method](#)), 14  
[JoinRequest](#) ([amino.lib.util.objects.JoinRequest attribute](#)), 38  
[JoinRequest](#) ([class in amino.lib.util.objects](#)), 38

## K

[kick\(\)](#) ([amino.client.Client method](#)), 14  
[kick\(\)](#) ([amino.sub\\_client.SubClient method](#)), 30

## L

[leave\\_chat\(\)](#) ([amino.client.Client method](#)), 15  
[leave\\_chat\(\)](#) ([amino.sub\\_client.SubClient method](#)), 30  
[leave\\_community\(\)](#) ([amino.client.Client method](#)), 15  
[LevelFiveRequiredToEnableProps](#), 48  
[LibraryUpdateAvailable](#), 48  
[like\\_blog\(\)](#) ([amino.client.Client method](#)), 15  
[like\\_blog\(\)](#) ([amino.sub\\_client.SubClient method](#)), 30  
[like\\_comment\(\)](#) ([amino.client.Client method](#)), 15  
[like\\_comment\(\)](#) ([amino.sub\\_client.SubClient method](#)), 30  
[link\\_identify\(\)](#) ([amino.client.Client method](#)), 15  
[list\\_communities\(\)](#) ([amino.acm.ACM method](#)), 3  
[LiveLayer](#) ([amino.lib.util.objects.LiveLayer attribute](#)), 38  
[LiveLayer](#) ([class in amino.lib.util.objects](#)), 38  
[login\(\)](#) ([amino.client.Client method](#)), 15  
[login\\_sid\(\)](#) ([amino.client.Client method](#)), 16  
[logout\(\)](#) ([amino.client.Client method](#)), 16  
[lottery\(\)](#) ([amino.sub\\_client.SubClient method](#)), 31  
[LotteryLog](#) ([amino.lib.util.objects.LotteryLog attribute](#)), 38  
[LotteryLog](#) ([class in amino.lib.util.objects](#)), 38



## M

`mark_as_read()` (*amino.client.Client method*), 16  
`mark_as_read()` (*amino.sub\_client.SubClient method*), 31  
`MemberKickedByOrganizer`, 48  
`Membership` (*amino.lib.util.objects.Membership attribute*), 38  
`Membership` (*class in amino.lib.util.objects*), 38  
`Message` (*amino.lib.util.objects.Message attribute*), 38  
`Message` (*class in amino.lib.util.objects*), 38  
`MessageList` (*amino.lib.util.objects.MessageList attribute*), 38  
`MessageList` (*class in amino.lib.util.objects*), 38  
`MessageNeeded`, 49  
`moderation_history()` (*amino.sub\_client.SubClient method*), 31

## N

`news_feed` (*amino.lib.util.objects.Objects.Users attribute*), 38  
`NoChatThread`, 49  
`NoCommunity`, 49  
`NoImageSource`, 49  
`NotEnoughCoins`, 49  
`NoticeList` (*amino.lib.util.objects.NoticeList attribute*), 38  
`NoticeList` (*class in amino.lib.util.objects*), 38  
`NotificationList` (*amino.lib.util.objects.NotificationList attribute*), 38  
`NotificationList` (*class in amino.lib.util.objects*), 38  
`NotLoggedIn`, 49  
`NotOwnerOfChatBubble`, 49

## O

`Objects` (*class in amino.lib.util.objects*), 38  
`Objects.Users` (*class in amino.lib.util.objects*), 38  
`on_avatar_chat_end()` (*amino.socket.Callbacks method*), 34  
`on_avatar_chat_not_answered()` (*amino.socket.Callbacks method*), 34  
`on_avatar_chat_not_cancelled()` (*amino.socket.Callbacks method*), 34  
`on_avatar_chat_not_declined()` (*amino.socket.Callbacks method*), 34  
`on_avatar_chat_start()` (*amino.socket.Callbacks method*), 34  
`on_chat_background_changed()` (*amino.socket.Callbacks method*), 34  
`on_chat_content_changed()` (*amino.socket.Callbacks method*), 34  
`on_chat_host_transferred()` (*amino.socket.Callbacks method*), 34

`on_chat_icon_changed()` (*amino.socket.Callbacks method*), 34  
`on_chat_invite()` (*amino.socket.Callbacks method*), 34  
`on_chat_pin_announcement()` (*amino.socket.Callbacks method*), 34  
`on_chat_removed_message()` (*amino.socket.Callbacks method*), 34  
`on_chat_tip()` (*amino.socket.Callbacks method*), 34  
`on_chat_tipping_disabled()` (*amino.socket.Callbacks method*), 34  
`on_chat_tipping_enabled()` (*amino.socket.Callbacks method*), 34  
`on_chat_title_changed()` (*amino.socket.Callbacks method*), 34  
`on_chat_unpin_announcement()` (*amino.socket.Callbacks method*), 34  
`on_chat_view_only_disabled()` (*amino.socket.Callbacks method*), 34  
`on_chat_view_only_enabled()` (*amino.socket.Callbacks method*), 35  
`on_delete_message()` (*amino.socket.Callbacks method*), 35  
`on_group_member_join()` (*amino.socket.Callbacks method*), 35  
`on_group_member_leave()` (*amino.socket.Callbacks method*), 35  
`on_image_message()` (*amino.socket.Callbacks method*), 35  
`on_invite_message()` (*amino.socket.Callbacks method*), 35  
`on_screen_room_end()` (*amino.socket.Callbacks method*), 35  
`on_screen_room_start()` (*amino.socket.Callbacks method*), 35  
`on_sticker_message()` (*amino.socket.Callbacks method*), 35  
`on_strike_message()` (*amino.socket.Callbacks method*), 35  
`on_text_message()` (*amino.socket.Callbacks method*), 35  
`on_text_message_force_removed()` (*amino.socket.Callbacks method*), 35  
`on_text_message_removed_by_admin()` (*amino.socket.Callbacks method*), 35  
`on_timestamp_message()` (*amino.socket.Callbacks method*), 35  
`on_user_typing_end()` (*amino.socket.Callbacks method*), 35  
`on_user_typing_start()` (*amino.socket.Callbacks method*), 35  
`on_video_chat_end()` (*amino.socket.Callbacks method*), 35  
`on_video_chat_not_answered()`

(amino.socket.Callbacks method), 35  
 on\_video\_chat\_not\_cancelled() (amino.socket.Callbacks method), 35  
 on\_video\_chat\_not\_declined() (amino.socket.Callbacks method), 35  
 on\_video\_chat\_start() (amino.socket.Callbacks method), 35  
 on\_voice\_chat\_end() (amino.socket.Callbacks method), 35  
 on\_voice\_chat\_not\_answered() (amino.socket.Callbacks method), 35  
 on\_voice\_chat\_not\_cancelled() (amino.socket.Callbacks method), 35  
 on\_voice\_chat\_not\_declined() (amino.socket.Callbacks method), 35  
 on\_voice\_chat\_permission\_invite\_only() (amino.socket.Callbacks method), 35  
 on\_voice\_chat\_permission\_invited\_and\_requested() (amino.socket.Callbacks method), 35  
 on\_voice\_chat\_permission\_open\_to\_everyone() (amino.socket.Callbacks method), 35  
 on\_voice\_chat\_start() (amino.socket.Callbacks method), 35  
 on\_voice\_message() (amino.socket.Callbacks method), 35  
 on\_welcome\_message() (amino.socket.Callbacks method), 35  
 on\_youtube\_message() (amino.socket.Callbacks method), 35

## P

PageRepostedTooRecently, 49  
 parse\_headers() (amino.acm.ACM method), 4  
 parse\_headers() (amino.client.Client method), 16  
 parse\_headers() (amino.sub\_client.SubClient method), 31  
 play\_quiz() (amino.sub\_client.SubClient method), 31  
 play\_quiz\_raw() (amino.sub\_client.SubClient method), 31  
 post\_blog() (amino.sub\_client.SubClient method), 31  
 post\_wiki() (amino.sub\_client.SubClient method), 31  
 promote() (amino.acm.ACM method), 4  
 promotion() (amino.sub\_client.SubClient method), 31  
 purchase() (amino.sub\_client.SubClient method), 31

## Q

QuizAnswers (amino.lib.util.objects.QuizAnswers attribute), 38  
 QuizAnswers (class in amino.lib.util.objects), 38

QuizQuestionList (amino.lib.util.objects.QuizQuestionList attribute), 38  
 QuizQuestionList (class in amino.lib.util.objects), 38  
 QuizRanking (amino.lib.util.objects.QuizRanking attribute), 38  
 QuizRanking (class in amino.lib.util.objects), 38  
 QuizRankingList (amino.lib.util.objects.QuizRankingList attribute), 39  
 QuizRankingList (class in amino.lib.util.objects), 38  
 QuizRankings (amino.lib.util.objects.QuizRankings attribute), 39  
 QuizRankings (class in amino.lib.util.objects), 39

## R

RankingTableList (amino.lib.util.objects.RankingTableList attribute), 39  
 RankingTableList (class in amino.lib.util.objects), 39  
 ReachedMaxCategories, 49  
 ReachedMaxPollOptions, 50  
 ReachedMaxTitles, 50  
 ReachedTitleLength, 50  
 ReasonNeeded, 50  
 RecentBlogs (amino.lib.util.objects.RecentBlogs attribute), 39  
 RecentBlogs (class in amino.lib.util.objects), 39  
 register() (amino.client.Client method), 16  
 reject\_join\_request() (amino.acm.ACM method), 4  
 reject\_wiki\_request() (amino.sub\_client.SubClient method), 31  
 remove\_influencer() (amino.acm.ACM method), 4  
 remove\_linked\_community() (amino.client.Client method), 16  
 RemovedFromChat, 50  
 reorder\_featured\_users() (amino.sub\_client.SubClient method), 31  
 reorder\_linked\_communities() (amino.client.Client method), 17  
 repair\_check\_in() (amino.sub\_client.SubClient method), 31  
 reply\_wall() (amino.sub\_client.SubClient method), 31  
 repost\_blog() (amino.sub\_client.SubClient method), 31  
 request\_join\_community() (amino.client.Client method), 17  
 request\_verify\_code() (amino.client.Client method), 17  
 RequestedNoLongerExists, 50  
 RequestRejected, 50  
 resolve() (amino.socket.Callbacks method), 35

`restore()` (*amino.client.Client* method), 17  
`review_quiz_questions()`  
    (*amino.sub\_client.SubClient* method), 31  
`run()` (*amino.socket.SocketHandler* method), 36  
`run_amino_socket()` (*amino.socket.SocketHandler*  
    method), 36  
`run_vc()` (*amino.client.Client* method), 17

## S

`search_community()` (*amino.client.Client* method),  
    17  
`search_users()` (*amino.sub\_client.SubClient*  
    method), 31  
`send()` (*amino.socket.SocketHandler* method), 36  
`send_action()` (*amino.sub\_client.SubClient*  
    method), 31  
`send_coins()` (*amino.client.Client* method), 18  
`send_coins()` (*amino.sub\_client.SubClient* method),  
    31  
`send_message()` (*amino.client.Client* method), 18  
`send_message()` (*amino.sub\_client.SubClient*  
    method), 31  
`ServiceUnderMaintenance`, 50  
`set_amino_id()` (*amino.client.Client* method), 18  
`set_privacy_status()` (*amino.client.Client*  
    method), 18  
`SharedFolderFile` (*amino.lib.util.objects.SharedFolderFile*  
    attribute), 39  
`SharedFolderFile` (class in *amino.lib.util.objects*),  
    39  
`SharedFolderFileList`  
    (*amino.lib.util.objects.SharedFolderFileList*  
    attribute), 39  
`SharedFolderFileList` (class in  
    *amino.lib.util.objects*), 39  
`sid_to_ip_address()` (in module  
    *amino.lib.util.helpers*), 53  
`sid_to_uid()` (in module *amino.lib.util.helpers*), 53  
`SocketHandler` (class in *amino.socket*), 35  
`SpecifyType`, 51  
`start_chat()` (*amino.client.Client* method), 19  
`start_chat()` (*amino.sub\_client.SubClient* method),  
    32  
`start_vc()` (*amino.client.Client* method), 19  
`startup()` (*amino.socket.SocketHandler* method), 36  
`Sticker` (*amino.lib.util.objects.Sticker* attribute), 39  
`Sticker` (class in *amino.lib.util.objects*), 39  
`StickerCollection`  
    (*amino.lib.util.objects.StickerCollection* at-  
    tribute), 39  
`StickerCollection` (class in *amino.lib.util.objects*),  
    39  
`StickerCollectionList`  
    (*amino.lib.util.objects.StickerCollectionList*

    attribute), 39  
`StickerCollectionList` (class in  
    *amino.lib.util.objects*), 39  
`StickerList` (*amino.lib.util.objects.StickerList*  
    attribute), 39  
`StickerList` (class in *amino.lib.util.objects*), 39  
`strike()` (*amino.sub\_client.SubClient* method), 32  
`sub_clients()` (*amino.client.Client* method), 19  
`sub_clients_profile()` (*amino.client.Client*  
    method), 19  
`SubClient` (class in *amino.sub\_client*), 21  
`submit_to_wiki()` (*amino.sub\_client.SubClient*  
    method), 32  
`subscribe()` (*amino.sub\_client.SubClient* method),  
    32  
`subscribe_amino_plus()` (*amino.client.Client*  
    method), 19

## T

`team_amino` (*amino.lib.util.objects.Objects.Users* at-  
    tribute), 38  
`thank_tip()` (*amino.sub\_client.SubClient* method),  
    32  
`Thread` (*amino.lib.util.objects.Thread* attribute), 39  
`Thread` (class in *amino.lib.util.objects*), 39  
`ThreadList` (*amino.lib.util.objects.ThreadList* at-  
    tribute), 39  
`ThreadList` (class in *amino.lib.util.objects*), 39  
`TippedUsersSummary`  
    (*amino.lib.util.objects.TippedUsersSummary*  
    attribute), 39  
`TippedUsersSummary` (class in  
    *amino.lib.util.objects*), 39  
`TooManyChats`, 51  
`TooManyInviteUsers`, 51  
`TooManyRequests`, 51  
`transfer_host()` (*amino.sub\_client.SubClient*  
    method), 32  
`transfer_organizer()`  
    (*amino.sub\_client.SubClient* method), 32  
`TransferRequestNeeded`, 51

## U

`unban()` (*amino.sub\_client.SubClient* method), 32  
`unblock()` (*amino.client.Client* method), 19  
`unblock()` (*amino.sub\_client.SubClient* method), 33  
`UnexistentData`, 51  
`unfeature()` (*amino.sub\_client.SubClient* method),  
    33  
`unfollow()` (*amino.client.Client* method), 20  
`unfollow()` (*amino.sub\_client.SubClient* method), 33  
`unhide()` (*amino.sub\_client.SubClient* method), 33  
`UnknownResponse`, 51  
`unlike_blog()` (*amino.client.Client* method), 20



unlike\_blog() (amino.sub\_client.SubClient method), 33  
 unlike\_comment() (amino.client.Client method), 20  
 unlike\_comment() (amino.sub\_client.SubClient method), 33  
 UnsupportedEmail, 51  
 UnsupportedLanguage, 51  
 UnsupportedService, 51  
 unvote\_comment() (amino.sub\_client.SubClient method), 33  
 upload\_media() (amino.client.Client method), 20  
 upload\_themepack\_raw() (amino.acm.ACM method), 4  
 upvote\_comment() (amino.sub\_client.SubClient method), 33  
 UserAchievements (amino.lib.util.objects.UserAchievements attribute), 39  
 UserAchievements (class in amino.lib.util.objects), 39  
 UserBannedByTeamAmino, 52  
 UserCheckIns (amino.lib.util.objects.UserCheckIns attribute), 40  
 UserCheckIns (class in amino.lib.util.objects), 40  
 UserNotJoined, 52  
 UserNotMemberOfCommunity, 52  
 UserProfile (amino.lib.util.objects.UserProfile attribute), 40  
 UserProfile (class in amino.lib.util.objects), 40  
 UserProfileCountList (amino.lib.util.objects.UserProfileCountList attribute), 40  
 UserProfileCountList (class in amino.lib.util.objects), 40  
 UserProfileList (amino.lib.util.objects.UserProfileList attribute), 40  
 UserProfileList (class in amino.lib.util.objects), 40  
 UserSavedBlogs (amino.lib.util.objects.UserSavedBlogs attribute), 40  
 UserSavedBlogs (class in amino.lib.util.objects), 40  
 UserUnavailable, 52

## V

vc\_permission() (amino.sub\_client.SubClient method), 33  
 VCHeaders (class in amino.sub\_client), 34  
 VcReputation (amino.lib.util.objects.VcReputation attribute), 40  
 VcReputation (class in amino.lib.util.objects), 40  
 VerificationRequired, 52  
 verify() (amino.client.Client method), 21  
 visit() (amino.client.Client method), 21  
 visit() (amino.sub\_client.SubClient method), 34  
 VisitorsList (amino.lib.util.objects.VisitorsList attribute), 40  
 VisitorsList (class in amino.lib.util.objects), 40  
 vote\_poll() (amino.sub\_client.SubClient method), 34

## W

WallCommentingDisabled, 52  
 wallet\_config() (amino.client.Client method), 21  
 WalletHistory (amino.lib.util.objects.WalletHistory attribute), 40  
 WalletHistory (class in amino.lib.util.objects), 40  
 WalletInfo (amino.lib.util.objects.WalletInfo attribute), 40  
 WalletInfo (class in amino.lib.util.objects), 40  
 warn() (amino.sub\_client.SubClient method), 34  
 Wiki (amino.lib.util.objects.Wiki attribute), 40  
 Wiki (class in amino.lib.util.objects), 40  
 WikiCategory (amino.lib.util.objects.WikiCategory attribute), 40  
 WikiCategory (class in amino.lib.util.objects), 40  
 WikiCategoryList (amino.lib.util.objects.WikiCategoryList attribute), 40  
 WikiCategoryList (class in amino.lib.util.objects), 40  
 WikiLabelList (amino.lib.util.objects.WikiLabelList attribute), 40  
 WikiLabelList (class in amino.lib.util.objects), 40  
 WikiList (amino.lib.util.objects.WikiList attribute), 41  
 WikiList (class in amino.lib.util.objects), 40  
 WikiRequestList (amino.lib.util.objects.WikiRequestList attribute), 41  
 WikiRequestList (class in amino.lib.util.objects), 41  
 WrongType, 52

## Y

YouAreBanned, 52