Topgg Node SDK

Top.gg

Mar 18, 2021

CONTENTS

CHAPTER

ONE

INSTALLATION

You can install the official node sdk via

`npm i @top-gg/sdk` or `yarn add @top-gg/sdk`

CHAPTER

TWO

EXAMPLES

You can find examples on the NPM Package page

CHAPTER

THREE

CHANGE LOG

3.1 3.0.9

- A change to how typings are used, properly exports via dist/typings

3.2 3.0.8

- Change docs URL
- Fix typos in a few method docs
- Specified that .hasVoted() is last 12 hours
- Auto-respond with status 200 as per docs

3.3 3.0.7

- Add new BotInfo.bannerUrl to typings

3.4 3.0.6

- Fix to webhook, where there were issues when using a body parser that also used `raw-body`

3.5 3.0.5

- Fix to Api.isWeekend example

3.6 3.0.4

- Released official documentation page
- Added GitHub actions to publish to npm on release

3.7 3.0.3

- Fixed some webhook bugs
- Improved a few typings

3.8 3.0.2

Some day-one patches to README, 3.0.1 is a dud release

3.9 3.0.0

A full rewrite of the package, here's some of the major changes:

- The webhook and API interaction have been full separated, into two different classes, `Api` and `Webhook`
- Auto-Posting is no longer natively supported in the package, instead you can use the officially *supported* `topgg-autoposter` package.
- The webhook class is no longer ran off of events, instead acting as a middleware function that defines `req.vote`

CHAPTER FOUR

API

Top.gg API Client for Posting stats or Fetching data

Kind: global class**Link**: https://topggjs.rtfd.io <- Library docs**Link**: https://docs.top.gg <- API Reference

- *Api*
  - *new Api(token, options)*
  - *.postStats(stats)* BotStats
  - *.getStats(id)* BotStats
  - *.getBot(id)* BotInfo
  - *.getUser(id)* UserInfo
  - *.getBots(query)* BotsResponse
  - *.getVotes()* Array.<ShortUser>
  - *.hasVoted(id)* Boolean
  - *.isWeekend()* Boolean

4.1 new Api(token, options)

Create Top.gg API instance

Example

```
const Topgg = require(`@top-gg/sdk`)

const api = new Topgg.Api('Your top.gg token')
```

4.2 api.postStats(stats) BotStats

Post bot stats to Top.gg (Do not use if you supplied a client)

Kind: instance method of *Api***Returns**: BotStats - Passed object

Example

```
await client.postStats({
  serverCount: 28199,
  shardCount: 1
})
```

4.3 api.getStats(id) BotStats

Get a bots stats

Kind: instance method of *Api***Returns**: BotStats - Stats of bot requested

Example

```
await client.getStats('461521980492087297')
// =>
{
  serverCount: 28199,
  shardCount 1,
  shards: []
}
```

4.4 api.getBot(id) BotInfo

Get bot info

Kind: instance method of *Api***Returns**: BotInfo - Info for bot

Example

```
await client.getBot('461521980492087297') // returns bot info
```

4.5 api.getUser(id) UserInfo

Get user info

Kind: instance method of *Api***Returns**: UserInfo - Info for user

Example

```
await client.getUser('205680187394752512')
// =>
user.username // Xignotic
```

4.6 api.getBots(query) BotsResponse

Get a list of bots

Kind: instance method of *Api***Returns**: BotsResponse - Return response

Example

```
// Finding by properties
await client.getBots({
  search: {
    username: 'shiro',
    certifiedBot: true
    ...any other bot object properties
  }
})
// =>
{
  results: [
    {
      id: '461521980492087297',
      username: 'Shiro',
      discriminator: '8764',
      lib: 'discord.js',
      ...rest of bot object
    }
    ...other shiro knockoffs B)
  ],
  limit: 10,
  offset: 0,
  count: 1,
  total: 1
}
// Restricting fields
await client.getBots({
  fields: ['id', 'username']
})
// =>
{
  results: [
    {
      id: '461521980492087297',
```

(continues on next page)

(continued from previous page)

```
      username: 'Shiro'
    },
    {
      id: '493716749342998541',
      username: 'Mimu'
    },
    ...
  ],
  ...
}
```

4.7 api.getVotes() Array.<ShortUser>

Get users who've voted

Kind: instance method of *Api***Returns**: Array.<ShortUser> - Array of users who've voted**Example**

```
await client.getVotes()
// =>
[
  {
    username: 'Xignotic',
    discriminator: '0001',
    id: '205680187394752512',
    avatar: '3b9335670c7213b3a2d4e990081900c7'
  },
  {
    username: 'iara',
    discriminator: '0001',
    id: '395526710101278721',
    avatar: '3d1477390b8d7c3cec717ac5c778f5f4'
  }
  ...more
]
```

4.8 api.hasVoted(id) Boolean

Get whether or not a user has voted in the last 12 hours

Kind: instance method of *Api***Returns**: Boolean - Whether the user has voted in the last 12 hours

Example

```
await client.hasVoted('205680187394752512')
// => true/false
```

4.9 api.isWeekend() Boolean

Whether or not the weekend multiplier is active

Kind: instance method of *Api***Returns**: Boolean - Whether the multiplier is active**Example**

```
await client.isWeekend()
// => true/false
```

CHAPTER

FIVE

WEBHOOK

Top.gg Webhook

Kind: global class

- *Webhook*
  - *new Webhook(authorization)*
  - *.middleware()*

5.1 new Webhook(authorization)

Create a new webhook client instance

Example

```
const express = require('express')
const { Webhook } = require(`@top-gg/sdk`)

const app = express()
const wh = new Webhook('webhookauth123')

app.post('/dblwebhook', wh.middleware(), (req, res) => {
  // req.vote is your vote object e.g
  console.log(req.vote.user) // => 321714991050784770
})

app.listen(80)

// In this situation, your TopGG Webhook dashboard should look like
// URL = http://your.server.ip:80/dblwebhook
// Authorization: webhookauth123
```

5.2 webhook.middleware()

Middleware function to pass to express, sets req.vote to the payload

Kind: instance method of *Webhook***Example**

```
app.post('/dblwebhook', wh.middleware(), (req, res) => {
  // req.vote is your payload e.g
  console.log(req.vote.user) // => 395526710101278721
})
```