

```

package main

import (
    "context"
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gcron"
    "github.com/gogf/gf/v2/os/gctx"
    "time"
)

func main() {
    var (
        err error
        ctx = gctx.New()
    )
    _, err = gcron.Add(ctx, "* * * * *", func(ctx context.Context) {
        g.Log().Print(ctx, "Every second")
    }, "MySecondCronJob")
    if err != nil {
        panic(err)
    }

    _, err = gcron.Add(ctx, "0 30 * * * *", func(ctx context.Context) {
        g.Log().Print(ctx, "Every hour on the half hour")
    })
    if err != nil {
        panic(err)
    }

    _, err = gcron.Add(ctx, "@hourly", func(ctx context.Context) {
        g.Log().Print(ctx, "Every hour")
    })
    if err != nil {
        panic(err)
    }

    _, err = gcron.Add(ctx, "@every 1h30m", func(ctx context.Context) {
        g.Log().Print(ctx, "Every hour thirty")
    })
    if err != nil {
        panic(err)
    }

    g.Dump(gcron.Entries())

    time.Sleep(3 * time.Second)

    g.Log().Print(ctx, `stop cronjob "MySecondCronJob"`)
    gcron.Stop("MySecondCronJob")

    time.Sleep(3 * time.Second)

    g.Log().Print(ctx, `start cronjob "MySecondCronJob"`)
    gcron.Start("MySecondCronJob")

    time.Sleep(3 * time.Second)
}

```

Content Menu

-
-
-
-
-
-
-
-

```
[
  {
    Name: "MySecondCronJob",
    Job: 0x14077e0,
    Time: "2021-11-14 12:13:53.445132 +0800 CST m=+0.006167069",
  },
  {
    Name: "cron-1",
    Job: 0x14078a0,
    Time: "2021-11-14 12:13:53.44515 +0800 CST m=+0.006185688",
  },
  {
    Name: "cron-2",
    Job: 0x1407960,
    Time: "2021-11-14 12:13:53.445161 +0800 CST m=+0.006196483",
  },
  {
    Name: "cron-3",
    Job: 0x1407a20,
    Time: "2021-11-14 12:13:53.445218 +0800 CST m=+0.006252937",
  },
]
2021-11-14 12:13:54.442 {189cwi9ngk0cfp7l8gcwciw100sr9cuu} Every second
2021-11-14 12:13:55.441 {189cwi9ngk0cfp7l8gcwciw100sr9cuu} Every second
2021-11-14 12:13:56.440 {189cwi9ngk0cfp7l8gcwciw100sr9cuu} Every second
2021-11-14 12:13:56.445 {189cwi9ngk0cfp7l8gcwciw100sr9cuu} stop cronjob
"MySecondCronJob"
2021-11-14 12:13:59.445 {189cwi9ngk0cfp7l8gcwciw100sr9cuu} start cronjob
"MySecondCronJob"
2021-11-14 12:14:00.443 {189cwi9ngk0cfp7l8gcwciw100sr9cuu} Every second
2021-11-14 12:14:01.442 {189cwi9ngk0cfp7l8gcwciw100sr9cuu} Every second
2021-11-14 12:14:02.443 {189cwi9ngk0cfp7l8gcwciw100sr9cuu} Every second
```

AddSingleton

```
package main

import (
    "context"
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gcron"
    "github.com/gogf/gf/v2/os/gctx"
    "time"
)

func main() {
    var (
        err error
        ctx = gctx.New()
    )
    _, err = gcron.AddSingleton(ctx, " * * * * *", func(ctx context.Context) {
        g.Log().Print(ctx, "doing")
        time.Sleep(2 * time.Second)
    })
    if err != nil {
        panic(err)
    }
    select {}
}
```

```
2021-11-14 12:16:54.073 {189cwi9nmm0cfp7niz319fc100zrw0ig} doing
2021-11-14 12:16:57.072 {189cwi9nmm0cfp7niz319fc100zrw0ig} doing
2021-11-14 12:17:00.072 {189cwi9nmm0cfp7niz319fc100zrw0ig} doing
2021-11-14 12:17:03.071 {189cwi9nmm0cfp7niz319fc100zrw0ig} doing
2021-11-14 12:17:06.072 {189cwi9nmm0cfp7niz319fc100zrw0ig} doing
2021-11-14 12:17:09.072 {189cwi9nmm0cfp7niz319fc100zrw0ig} doing
...
```

AddOnceSize

```
func main() {
    var (
        ctx = gctx.New()
    )
    cron := gcron.New()
    array := garray.New(true)
    cron.AddOnce(ctx, "@every 2s", func(ctx context.Context) {
        array.Append(1)
    })
    fmt.Println(cron.Size(), array.Len())
    time.Sleep(3000 * time.Millisecond)
    fmt.Println(cron.Size(), array.Len())
}
```

```
1 0
0 1
```

AddTimesSize

```
func main() {
    var (
        ctx = gctx.New()
    )
    cron := gcron.New()
    array := garray.New(true)
    cron.AddTimes(ctx, "@every 2s", 2, func(ctx context.Context) {
        array.Append(1)
    })
    fmt.Println(cron.Size(), array.Len())
    time.Sleep(3000 * time.Millisecond)
    fmt.Println(cron.Size(), array.Len())
    time.Sleep(3000 * time.Millisecond)
    fmt.Println(cron.Size(), array.Len())
}
```

```
1 0
1 1
0 2
```

Entriesasc

```
func main() {
    var (
        ctx = gctx.New()
    )
    cron := gcron.New()
    array := garray.New(true)
    cron.AddTimes(ctx, "@every 1s", 2, func(ctx context.Context) {
        array.Append(1)
    }, "cron1")
    cron.AddOnce(ctx, "@every 1s", func(ctx context.Context) {
        array.Append(1)
    }, "cron2")
    entries := cron.Entries()
    for k, v := range entries {
        fmt.Println(k, v.Name, v.Time)
    }
    time.Sleep(3000 * time.Millisecond)
    fmt.Println(array.Len())
}
```

```
0 cron2 2022-02-09 10:11:47.2421345 +0800 CST m=+0.159116501
1 cron1 2022-02-09 10:11:47.2421345 +0800 CST m=+0.159116501
3
```

Search""(*Entry)nil

```
func main() {
    var (
        ctx = gctx.New()
    )
    cron := gcron.New()
    array := garray.New(true)
    cron.AddTimes(ctx, "@every 1s", 2, func(ctx context.Context) {
        array.Append(1)
    }, "cron1")
    cron.AddOnce(ctx, "@every 1s", func(ctx context.Context) {
        array.Append(1)
    }, "cron2")
    search := cron.Search("cron2")

    g.Log().Print(ctx, search)

    time.Sleep(3000 * time.Millisecond)
    fmt.Println(array.Len())

    // Output:
    // 3
}
```

```
2022-02-09 10:52:30.011 {18a909957cfed11680c1b145da1ef096} {"Name":"cron2","Time":"2022-02-09T10:52:29.9972842+08:00"}
```

`Stop`(`Stop`), `name``namecron`

```
func main() {
    var (
        ctx = gctx.New()
    )
    cron := gcron.New()
    array := garray.New(true)
    cron.AddTimes(ctx, "@every 2s", 1, func(ctx context.Context) {
        array.Append(1)
    }, "cron1")
    cron.AddOnce(ctx, "@every 2s", func(ctx context.Context) {
        array.Append(1)
    }, "cron2")
    fmt.Println(array.Len(), cron.Size())
    cron.Stop("cron2")
    fmt.Println(array.Len(), cron.Size())
    time.Sleep(3000 * time.Millisecond)
    fmt.Println(array.Len(), cron.Size())

    // Output:
    // 1
    // 1
}
```

```
0 2
0 2
1 1
```

`Remove``name()`

```

func main() {
    var (
        ctx = gctx.New()
    )
    cron := gcron.New()
    array := garray.New(true)
    cron.AddTimes(ctx, "@every 2s", 1, func(ctx context.Context) {
        array.Append(1)
    }, "cron1")
    cron.AddOnce(ctx, "@every 2s", func(ctx context.Context) {
        array.Append(1)
    }, "cron2")
    fmt.Println(array.Len(), cron.Size())
    cron.Remove("cron2")
    fmt.Println(array.Len(), cron.Size())
    time.Sleep(3000 * time.Millisecond)
    fmt.Println(array.Len(), cron.Size())
    // Output:
    // 0 2
    // 0 1
    // 1 0
}

```

```

0 2
0 1
1 0

```

Start(Add), name name cron

```

func main() {
    var (
        ctx = gctx.New()
    )
    cron := gcron.New()
    array := garray.New(true)
    cron.AddOnce(ctx, "@every 2s", func(ctx context.Context) {
        array.Append(1)
    }, "cron2")
    cron.Stop("cron2")
    time.Sleep(3000 * time.Millisecond)
    fmt.Println(array.Len(), cron.Size())
    cron.Start("cron2")
    time.Sleep(3000 * time.Millisecond)
    fmt.Println(array.Len(), cron.Size())

    // Output:
    // 0 1
    // 1 0
}

```

```

0 1
1 0

```

