

# concurrent

\* used by 7.2k projects go documentation build passing codecov 80% go report A

License Apache 2.0

- concurrent.Map: backport sync.Map for go below 1.9
- concurrent.Executor: goroutine with explicit ownership and cancellable

## concurrent.Map

because sync.Map is only available in go 1.9, we can use concurrent.Map to make code portable

```
m := concurrent.NewMap()
m.Store("hello", "world")
elem, found := m.Load("hello")
// elem will be "world"
// found will be true
```

## concurrent.Executor

```
executor := concurrent.NewUnboundedExecutor()
executor.Go(func(ctx context.Context) {
    everyMillisecond := time.NewTicker(time.Millisecond)
    for {
        select {
        case <-ctx.Done():
            fmt.Println("goroutine exited")
            return
        case <-everyMillisecond.C:
            // do something
        }
    }
})
time.Sleep(time.Second)
executor.StopAndWaitForever()
fmt.Println("executor stopped")
```

attach goroutine to executor instance, so that we can

- cancel it by stop the executor with Stop/StopAndWait/StopAndWaitForever
- handle panic by callback: the default behavior will no longer crash your application