

Go expvar Paketi Cheatsheet

expvar paketi, Go uygulamalarında çalışma sırasında istatistikleri ve değişkenleri HTTP üzerinden JSON formatında dışarıya açmaya yarar. Genellikle monitoring, debugging ve performans ölçümü için kullanılır. Varsayılan olarak `/debug/vars` endpointinde yayınlanır.

■ Temel Kavramlar

- **Var**: Tüm expvar değişkenlerinin uyguladığı arayüz.
- **Int**: Atomik integer sayaç.
- **Float**: Atomik float değer.
- **String**: Atomik string değer.
- **Map**: String key, Var value içeren map.
- **Func**: Dinamik değer üreten fonksiyon.

■ Fonksiyonlar

- **Publish(name, v)**: Yeni bir değişkeni global tabloya ekler.
- **Get(name)**: Kaydedilmiş değeri döner.
- **Do(func(KeyValue))**: Tüm değişkenler üzerinde döner.
- **NewInt / NewFloat / NewString / NewMap / NewFunc**: Yeni değişken oluşturur ve tabloya ekler.

■ Kod Örnekleri

Int Sayaç Örneği:

```
var requests = expvar.NewInt("requests")

func handler(w http.ResponseWriter, r *http.Request) {
    requests.Add(1)
    fmt.Fprintln(w, "Hello, World!")
}
```

Map Kullanımı:

```
var stats = expvar.NewMap("stats")

func handler(w http.ResponseWriter, r *http.Request) {
    stats.Add("hits", 1)
    stats.Add("errors", 2)
}
```

Dinamik Func Örneği:

```
expvar.Publish("uptime", expvar.Func(func() interface{} {
    return time.Since(start).String()
}))
```

■ Otomatik Değerler

- **cmdline**: Programın argümanları.
- **memstats**: runtime.MemStats bilgileri (heap, GC, allocation vb.).

■ Tam Uygulama Örneği

```
var (
    requests = expvar.NewInt("requests")
    errors    = expvar.NewInt("errors")
    version   = expvar.NewString("version")
    stats     = expvar.NewMap("stats")
    start     = time.Now()
)

func main() {
    version.Set("1.0.0")
    expvar.Publish("uptime", expvar.Func(func() interface{} {
        return time.Since(start).String()
    }))

    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        requests.Add(1)
        stats.Add("hits", 1)
        fmt.Fprintln(w, "Hello, World!")
    })

    http.HandleFunc("/error", func(w http.ResponseWriter, r *http.Request) {
        errors.Add(1)
        stats.Add("errors", 1)
        http.Error(w, "Something went wrong!", 500)
    })

    http.ListenAndServe(":8080", nil)
}
```