

Mão na Massa: Testando múltiplos sites

Começando deste ponto? Você pode fazer o [DOWNLOAD \(https://s3.amazonaws.com/caelum-online-public/624-golang/04/projetos/alura-golang-stage-fim-cap04.zip\)](https://s3.amazonaws.com/caelum-online-public/624-golang/04/projetos/alura-golang-stage-fim-cap04.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

Neste exercício vamos melhorar o monitoramento de nosso programa, para que ele monitore mais de um site ao mesmo tempo.

1- Nosso primeiro passo é substituir a nossa string `site` pelo slice `sites`, que conterá os endereços dos vários sites a serem testados. Vamos testar pelo menos 3 sites, o da Alura (<https://www.alura.com.br>) , o da Caelum (<https://www.caelum.com.br>) e o último do Random Status Code (<https://random-status-code.herokuapp.com>) , para ter um falhe de vez em quando. Crie o slice com estes 3 sites:

```
//hello.go

//restante do arquivo

func iniciarMonitoramento() {
    fmt.Println("Monitorando...")

    sites := []string{"https://random-status-code.herokuapp.com/",
        "https://www.alura.com.br", "https://www.caelum.com.br"}
    //restante da função
}
```

2- Como queremos que cada um destes sites seja testado uma vez, vamos um for para percorrer o slice inteiro, utilize o range para facilitar:

```
//hello.go

//restante do arquivo

func iniciarMonitoramento() {
    fmt.Println("Monitorando...")

    sites := []string{"https://random-status-code.herokuapp.com/",
        "https://www.alura.com.br", "https://www.caelum.com.br"}

    for i, site := range sites {
        // restante
    }
}
```

3- Vamos agora extrair o nosso código que testa um site para uma função externa, para facilitar na hora que estamos iterando sobre o slice, aonde cada site deve executar uma vez a função. Crie a função `testasite` que deve receber uma string com o site a ser testado e move o código go `http.Get` para lá:

```

//hello.go
//restante do arquivo

func testaSite(site string) {

    resp, _ := http.Get(site)

    if resp.StatusCode == 200 {
        fmt.Println("Site:", site, "foi carregado com sucesso!")
    } else {
        fmt.Println("Site:", site, "está com problemas. Status Code:", resp.StatusCode)
    }
}

```

4- Vamos agora chamar a testa site dentro do nosso for que está varrendo o slice:

```

// restante do código omitido

func iniciarMonitoramento() {
    fmt.Println("Monitorando...")

    sites := []string{"https://random-status-code.herokuapp.com/",
        "https://www.alura.com.br", "https://www.caelum.com.br"}

    for i, site := range sites {
        fmt.Println("Testando site", i, ":", site)
        testaSite(site)
    }

    fmt.Println("")
}

```

_ Adicione o `fmt.Println("")` vazio ao final, e o que está dentro do for também, para que o nosso script vá exibindo mensagens para o nosso usuário enquanto ele é executado._

Execute o nosso script e veja que agora conseguimos testar diversos sites, e ao final do teste o nosso Menu surge para escolhermos a opção de monitorar novamente.