

Detectando erros

Transcrição

Para tratar os erros, primeiramente devemos substituir o operador de identificador em branco (_) pela variável `err` , padrão para os erros em Go:

```
// restante do código omitido

func leSitesDoArquivo() []string {

    var sites []string
    arquivo, err := os.Open("sites.txt")
    fmt.Println(arquivo)

    return sites
}
```

E agora, se `err` não for nulo, significa que houve um erro, então vamos imprimi-lo:

```
// restante do código omitido

func leSitesDoArquivo() []string {

    var sites []string
    arquivo, err := os.Open("sites.txt")

    if err != nil {
        fmt.Println("Ocorreu um erro:", err)
    }

    fmt.Println(arquivo)

    return sites
}
```

Agora, ao executar o programa e iniciar o monitoramento, vemos a seguinte impressão:

```
Ocorreu um erro: open sites.txt: no such file or directory
```

Agora está bem claro o que aconteceu, que o arquivo, ou o diretório, não foi encontrado. Agora, vamos tratar do outro erro que estamos ignorando, no momento em que fazemos a requisição para o site, na função `testasite` :

```
// restante do código omitido

func testasite(site string) {

    resp, err := http.Get(site)
```

```
if err != nil {
    fmt.Println("Ocorreu um erro:", err)
}

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

Agora que já estamos detectando os erros do programa, vamos prosseguir com a leitura do arquivo.