08

Mãos na Massa: Lendo sites do arquivo

> Começando deste ponto? Você pode fazer o DOWNLOAD (https://s3.amazonaws.com/caelum-online-public/624-golang/05/projetos/alura-golang-stage-fim-cap05.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

Neste exercício vamos fazer com que os sites monitorados venham de um arquivo `.txt` para que fique fácil adicionar ou remover um site do monitoramento, além disto vamos colocar algumas detecções de erro em nossas funções.

1- Primeiro, vamos colocar nosso aprendizado sobre erros deste capítulo para capturar os erros, o primeiro deles de uma função que já utilizamos, a `http.Get()` da funçao `testaSite` :

```
// restante do código omitido

func testaSite(site string) {

    //Capturando o segundo parâmetro
    resp, err := http.Get(site)
    //Verificando se houve algum erro
    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)
    }
}
```

2- Agora vamos começar a ler os nossos sites de um arquivo `.txt` . Crie um arquivo de texto chamado `sites.txt` e coloque os sites que você quer monitorar lá, um em cada linha:

sites.txt

```
https://www.alura.com.br
https://random-status-code.herokuapp.com
https://www.caelum.com.br
https://www.casadocodigo.com.br
```

3- Para ler este arquivo, vamos criar a função `leSitesDoArquivo` , que vai retornar para o nosso slice de sites preenchido de acordo com os sites do arquivo `sites.txt` . Crie a função `leSitesDoArquivo` que vai retornar um slice de strings:

```
func leSitesDoArquivo() []string {

    var sites []string
```

```
    return sites
}
```

4- Agora vamos abrir o arquivo de `sites.txt` e detectar caso algum erro aconteça na abertura. Vamos utilizar o pacote `os` para abrir com a função `Open:

```
func leSitesDoArquivo() []string {

    var sites []string

    arquivo, err := os.Open("sites.txt")

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


    arquivo.Close()
    return sites
}
```

Não esqueça de fechar o arquivo no final com a função `Close()` do arquivo

5- Agora vamos criar um leitor com o pacote `bufio` , para que facilite o processo de percorrer o arquivo `sites.txt` :

```
func leSitesDoArquivo() []string {

    var sites []string

    arquivo, err := os.Open("sites.txt")

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

    leitor := bufio.NewReader(arquivo)


    arquivo.Close()
    return sites
}
```

6- Agora vamos utilizar um `for` e ler linha a linha até que o leitor encontre o `EOF` , o que nos dará um erro e utilizaremos a instrução `break` para sair do loop, indicando que chegamos ao final. Para ler cada linha utilizaremos a função `ReadString` do leitor, lendo até o caractere `\n` que indica o final da linha:

```
func leSitesDoArquivo() []string {

    var sites []string

    arquivo, err := os.Open("sites.txt")
```

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

    leitor := bufio.NewReader(arquivo)
    for {
        linha, err := leitor.ReadString('\n')


        if err == io.EOF {
            break
        }
    }

    arquivo.Close()
    return sites
}
```

7- Por último vamos dar um trim em cada linha lida para remover caracteres especiais, como `\n` , espaços e tabs. E claro, após isto vamos adicionar a linha ao slice de sites:

```
func leSitesDoArquivo() []string {

    var sites []string

    arquivo, err := os.Open("sites.txt")

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

    leitor := bufio.NewReader(arquivo)
    for {
        linha, err := leitor.ReadString('\n')
        linha = strings.TrimSpace(linha)

        sites = append(sites, linha)

        if err == io.EOF {
            break
        }

    }

    arquivo.Close()
    return sites
}
```

8- Agora vamos chamar a nossa função récem criada dentro função `iniciarMonitoramento` , para que ao invés de obtermos os sites de um slice fixo, obteremos do arquivo `sites.txt` :

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

```
        sites := leSitesDoArquivo()

        // restante da função
    }
```

Faça o teste, adicione novos sites ao arquivos `sites.txt` e verifique se os sites estão sendos lidos e testados corretamente.