

# Package goscanner

```
import "goscanner"
```

[Overview](#)[Index](#)[Examples](#)[Subdirectories](#)

## Overview ▾

Package goscanner verifica se esistono porte in ascolto su host di rete.

## Index ▾

```
func CheckPortCtx(ctx context.Context, r CheckRequest) error  
type CheckRequest
```

## Examples (Expand All)

[CheckPortCtx](#)

## Package files

[goscanner.go](#)

## func CheckPortCtx

```
func CheckPortCtx(ctx context.Context, r CheckRequest) error
```

CheckPortCtx avvia la verifica sui parametri passati in r.

### ▾ Example

ExampleCheckPortCtx è un esempio di utilizzo del package goscanner.

Code:

```
ctx := context.TODO()  
  
openPort := goscanner.CheckRequest{  
    Protocol: "tcp",  
    Host:     "127.0.0.1",  
    Port:     "1123",  
}
```

```
closedPort := goscanner.CheckRequest{
    Protocol: "tcp",
    Host:     "127.0.0.1",
    Port:     "1124",
}

err1 := goscanner.CheckPortCtx(ctx, openPort)
err2 := goscanner.CheckPortCtx(ctx, closedPort)

fmt.Println(err1)
fmt.Println(err2)
```

Output:

```
<nil>
porta 1124 chiusa
```

## type CheckRequest

CheckRequest è la struttura per creare una richiesta di verifica di porta logica Port su indirizzo Host tramite Protocol in modo di Timeout secondi.

```
type CheckRequest struct {
    Protocol string
    Host     string
    Port     string
    Timeout  int
}
```

## Subdirectories

### Name

..

[cmd](#)

Build version go1.13.8.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)