

[Discover Packages](#) > [Standard library](#) > [net](#) > [http](#) > [cgi](#) **cgi**

package

standard library

Version: [go1.20.1](#) **Latest** | Published: Feb 14, 2023 | License: [BSD-3-Clause](#) | Imports: [18](#) |Imported by: [395](#)**Details**[✓ Valid go.mod file](#)  [✓ Redistributable license](#)  [✓ Tagged version](#) [✓ Stable version](#) [Learn more](#)**Repository**[cs.opensource.google/go/go](#)**Links**[🛡 Report a Vulnerability](#) Documentation <> **Documentation****Overview**

Package `cgi` implements CGI (Common Gateway Interface) as specified in [RFC 3875](#).

Note that using CGI means starting a new process to handle each request, which is typically less efficient than using a long-running server. This package is intended primarily for compatibility with existing systems.

Index`func Request() (*http.Request, error)``func RequestFromMap(params map[string]string) (*http.Request, error)``func Serve(handler http.Handler) error``type Handler``func (h *Handler) ServeHTTP(rw http.ResponseWriter, req *http.Request)`**Constants**

This section is empty.

Variables

This section is empty.

Functions

func Request

```
func Request() (*http.Request, error)
```

Request returns the HTTP request as represented in the current environment. This assumes the current program is being run by a web server in a CGI environment. The returned Request's Body is populated, if applicable.

func RequestFromMap

```
func RequestFromMap(params map[string]string) (*http.Request, error)
```

RequestFromMap creates an http.Request from CGI variables. The returned Request's Body field is not populated.

func Serve

```
func Serve(handler http.Handler) error
```

Serve executes the provided Handler on the currently active CGI request, if any. If there's no current CGI environment an error is returned. The provided handler may be nil to use http.DefaultServeMux.

Types

type Handler

```
type Handler struct {
    Path string // path to the CGI executable
    Root string // root URI prefix of handler or empty for "/"

    // Dir specifies the CGI executable's working directory.
    // If Dir is empty, the base directory of Path is used.
    // If Path has no base directory, the current working
    // directory is used.
    Dir string

    Env []string // extra environment variables to set, if any, as "key=value"
    InheritEnv []string // environment variables to inherit from host, as "key"
    Logger *log.Logger // optional log for errors or nil to use log.Print
    Args []string // optional arguments to pass to child process
    Stderr io.Writer // optional stderr for the child process; nil means os.Stderr

    // PathLocationHandler specifies the root http Handler that
    // should handle internal redirects when the CGI process
    // returns a Location header value starting with a "/", as
    // specified in RFC 3875 § 6.3.2. This will likely be
    // http.DefaultServeMux.
    //
    // If nil, a CGI response with a local URI path is instead sent
```

```
// back to the client and not redirected internally.  
PathLocationHandler http.Handler  
}
```

Handler runs an executable in a subprocess with a CGI environment.

func (*Handler) ServeHTTP

```
func (h *Handler) ServeHTTP(rw http.ResponseWriter, req *http.Request)
```

Source Files

[View all](#) 

[child.go](#)

[host.go](#)

Why Go

[Use Cases](#)

[Case Studies](#)

Get Started

[Playground](#)

[Tour](#)

[Stack Overflow](#)

[Help](#)

Packages

[Standard Library](#)

[About Go Packages](#)

About

[Download](#)

[Blog](#)

[Issue Tracker](#)

[Release Notes](#)

[Brand Guidelines](#)

[Code of Conduct](#)

Connect

[Twitter](#)

[GitHub](#)

[Slack](#)

[r/golang](#)

[Meetup](#)

[Golang Weekly](#)

[Copyright](#)

[Terms of Service](#)

[Privacy Policy](#)

[Report an Issue](#)



Google