

# Package: goserveR (via r-universe)

October 16, 2026

**Type** Package

**Title** HTTP(S) File Server with Range Requests and Unbounded CORS Using 'go'

**Version** 0.1.3-0.9000

**Description** Provides a lightweight HTTP(S) file server implemented in Go and exposed to R. It supports range requests, configurable CORS and COOP headers, and optional TLS. The package builds and runs a background server that can be cleanly shut down from R using an OS pipe and C/Go interop. Asynchronous log handling is also supported.

**License** GPL (>= 3)

**Depends** R (>= 4.4.0)

**Imports** utils

**SystemRequirements** Go (<https://golang.org>), GNU make

**URL** <https://github.com/souunkou-bioinfo/goServeR>

**BugReports** <https://github.com/souunkou-bioinfo/goServeR/issues>

**Encoding** UTF-8

**Suggests** curl, nanonext, tinytest

**Config/roxygen2/version** 8.0.0

**Config/pak/sysreqs** make

**Repository** <https://souunkou-bioinfo.r-universe.dev>

**Date/Publication** 2026-07-21 20:11:06 UTC

**RemoteUrl** <https://github.com/souunkou-bioinfo/goServeR>

**RemoteRef** HEAD

**RemoteSha** 51683fc9723ee50b65a1a744557b2fce4959ffd1

Contents

|                                       |           |
|---------------------------------------|-----------|
| .create_default_log_handler . . . . . | 2         |
| .default_log_callback . . . . .       | 3         |
| addAuthKey . . . . .                  | 3         |
| as.data.frame.server_list . . . . .   | 4         |
| clearAuthKeys . . . . .               | 4         |
| createFileLogHandler . . . . .        | 5         |
| createSilentLogHandler . . . . .      | 5         |
| createTLSCertificate . . . . .        | 6         |
| isRunning . . . . .                   | 6         |
| listAuthKeys . . . . .                | 7         |
| listServers . . . . .                 | 7         |
| print.server_info . . . . .           | 8         |
| print.server_list . . . . .           | 8         |
| registerLogHandler . . . . .          | 9         |
| removeAuthKey . . . . .               | 9         |
| removeLogHandler . . . . .            | 10        |
| removeTLSCertificate . . . . .        | 10        |
| runServer . . . . .                   | 11        |
| shutdownServer . . . . .              | 13        |
| StartServer . . . . .                 | 13        |
| summary.server_list . . . . .         | 14        |
| updateAuthKey . . . . .               | 15        |
| <b>Index</b>                          | <b>16</b> |

---

|                                           |
|-------------------------------------------|
| .create_default_log_handler               |
| <i>Create default console log handler</i> |

---

Description

Create default console log handler

Usage

.create\_default\_log\_handler(fd)

Arguments

fd                      file descriptor for log pipe

Value

external pointer to log handler

---

*.default\_log\_callback*    *Default log callback - writes to console*

---

### **Description**

Default log callback - writes to console

### **Usage**

`.default_log_callback(handler, message, user)`

### **Arguments**

|         |                             |
|---------|-----------------------------|
| handler | external pointer to handler |
| message | log message                 |
| user    | user data (unused)          |

---

*addAuthKey*                      *Add Authentication Key*

---

### **Description**

Add an API key to the authentication system

### **Usage**

`addAuthKey(server_handle, key)`

### **Arguments**

|               |                                                            |
|---------------|------------------------------------------------------------|
| server_handle | External pointer from runServer(blocking=FALSE, auth=TRUE) |
| key           | Character string, the API key to add                       |

### **Value**

Invisible TRUE

```
as.data.frame.server_list
```

*Convert server list to data frame*

---

**Description**

Convert server list to data frame

**Usage**

```
## S3 method for class 'server_list'
as.data.frame(x, ...)
```

**Arguments**

|     |                                |
|-----|--------------------------------|
| x   | a server_list object           |
| ... | additional arguments (ignored) |

**Value**

a data.frame with server information

---

|               |                                      |
|---------------|--------------------------------------|
| clearAuthKeys | <i>Clear All Authentication Keys</i> |
|---------------|--------------------------------------|

---

**Description**

Remove all API keys from the authentication system

**Usage**

```
clearAuthKeys(server_handle)
```

**Arguments**

|               |                                                            |
|---------------|------------------------------------------------------------|
| server_handle | External pointer from runServer(blocking=FALSE, auth=TRUE) |
|---------------|------------------------------------------------------------|

**Value**

Invisible TRUE

---

|                      |                                |
|----------------------|--------------------------------|
| createFileLogHandler | <i>Create file log handler</i> |
|----------------------|--------------------------------|

---

**Description**

Create file log handler

**Usage**

```
createFileLogHandler(fd, logfile = tempfile("goserveR_", fileext = ".log"))
```

**Arguments**

|         |                              |
|---------|------------------------------|
| fd      | file descriptor for log pipe |
| logfile | path to log file             |

**Value**

external pointer to log handler

---

|                        |                                          |
|------------------------|------------------------------------------|
| createSilentLogHandler | <i>Create silent log handler (no-op)</i> |
|------------------------|------------------------------------------|

---

**Description**

Create silent log handler (no-op)

**Usage**

```
createSilentLogHandler(fd)
```

**Arguments**

|    |                              |
|----|------------------------------|
| fd | file descriptor for log pipe |
|----|------------------------------|

**Value**

external pointer to log handler

---

|                      |                                                 |
|----------------------|-------------------------------------------------|
| createTLSCertificate | <i>Generate TLS credentials with 'nanonext'</i> |
|----------------------|-------------------------------------------------|

---

### Description

Generates a self-signed X.509 certificate and private key as PEM using `nanonext::write_cert()`, then writes them to temporary files accepted by Go's HTTPS server. The private-key file is restricted to the current user on platforms that support Unix-style file modes.

### Usage

```
createTLSCertificate(
  cn = "127.0.0.1",
  valid = "20301231235959",
  directory = tempdir()
)
```

### Arguments

|                        |                                                                      |
|------------------------|----------------------------------------------------------------------|
| <code>cn</code>        | Certificate common name, usually the server host name or IP address. |
| <code>valid</code>     | Certificate expiry in <code>yyyymmddhhmmss</code> format.            |
| <code>directory</code> | Directory in which to create the temporary PEM files.                |

### Value

A `goserveR_tls_certificate` object containing the in-memory PEM values and the `certfile` and `keyfile` paths. Pass the object directly to the `tls` argument of `runServer()`.

---

|           |                                                                |
|-----------|----------------------------------------------------------------|
| isRunning | <i>isRunning</i> Check if a background server is still running |
|-----------|----------------------------------------------------------------|

---

### Description

`isRunning` Check if a background server is still running

### Usage

```
isRunning(handle)
```

### Arguments

|                     |                                                                     |
|---------------------|---------------------------------------------------------------------|
| <code>handle</code> | external pointer returned by <code>runServer(blocking=FALSE)</code> |
|---------------------|---------------------------------------------------------------------|

### Value

logical, TRUE if server is running, FALSE otherwise

Examples

```
## Not run:
h <- runServer(dir = ".", addr = "127.0.0.1:8080", blocking = FALSE)
isRunning(h) # TRUE
shutdownServer(h)
isRunning(h) # FALSE

## End(Not run)
```

---

|              |                          |
|--------------|--------------------------|
| listAuthKeys | List Authentication Keys |
|--------------|--------------------------|

---

Description

Get all current API keys in the authentication system

Usage

```
listAuthKeys(server_handle)
```

Arguments

server\_handle    External pointer from runServer(blocking=FALSE, auth=TRUE)

Value

Character vector of current API keys

---

|             |                                                                           |
|-------------|---------------------------------------------------------------------------|
| listServers | listServers List all running background servers with detailed information |
|-------------|---------------------------------------------------------------------------|

---

Description

listServers List all running background servers with detailed information

Usage

```
listServers()
```

Value

a server\_list S3 object containing server information

---

|                   |                                                        |
|-------------------|--------------------------------------------------------|
| print.server_info | <i>Print method for individual server_info objects</i> |
|-------------------|--------------------------------------------------------|

---

**Description**

Print method for individual server\_info objects

**Usage**

```
## S3 method for class 'server_info'  
print(x, index = NULL, ...)
```

**Arguments**

|       |                                 |
|-------|---------------------------------|
| x     | a server_info object            |
| index | server index number for display |
| ...   | additional arguments (ignored)  |

---

|                   |                                             |
|-------------------|---------------------------------------------|
| print.server_list | <i>Print method for server_list objects</i> |
|-------------------|---------------------------------------------|

---

**Description**

Print method for server\_list objects

**Usage**

```
## S3 method for class 'server_list'  
print(x, ...)
```

**Arguments**

|     |                                |
|-----|--------------------------------|
| x   | a server_list object           |
| ... | additional arguments (ignored) |

---

|                    |                                                     |
|--------------------|-----------------------------------------------------|
| registerLogHandler | <i>Register a log handler for a file descriptor</i> |
|--------------------|-----------------------------------------------------|

---

**Description**

Register a log handler for a file descriptor

**Usage**

```
registerLogHandler(fd, callback, user = NULL)
```

**Arguments**

|          |                                           |
|----------|-------------------------------------------|
| fd       | file descriptor to monitor                |
| callback | R function to call when data is available |
| user     | user data passed to callback              |

**Value**

external pointer to log handler

---

|               |                                  |
|---------------|----------------------------------|
| removeAuthKey | <i>Remove Authentication Key</i> |
|---------------|----------------------------------|

---

**Description**

Remove an API key from the authentication system

**Usage**

```
removeAuthKey(server_handle, key)
```

**Arguments**

|               |                                                            |
|---------------|------------------------------------------------------------|
| server_handle | External pointer from runServer(blocking=FALSE, auth=TRUE) |
| key           | Character string, the API key to remove                    |

**Value**

Invisible TRUE

---

|                  |                             |
|------------------|-----------------------------|
| removeLogHandler | <i>Remove a log handler</i> |
|------------------|-----------------------------|

---

**Description**

Remove a log handler

**Usage**

removeLogHandler(handler)

**Arguments**

handler                  external pointer to log handler

**Value**

logical indicating success

---

|                      |                                              |
|----------------------|----------------------------------------------|
| removeTLSCertificate | <i>Delete generated TLS credential files</i> |
|----------------------|----------------------------------------------|

---

**Description**

Delete generated TLS credential files

**Usage**

removeTLSCertificate(credentials)

**Arguments**

credentials              An object returned by createTLSCertificate().

**Value**

Invisibly, TRUE if neither credential file remains.

---

runServer

*runServer*


---

## Description

Run the go http server (blocking or background)

## Usage

```
runServer(
  dir = getwd(),
  addr = "0.0.0.0:8181",
  prefix = "",
  blocking = TRUE,
  cors = FALSE,
  coop = FALSE,
  tls = FALSE,
  certfile = "cert.pem",
  keyfile = "key.pem",
  silent = FALSE,
  log_handler = NULL,
  auth_keys = c(),
  auth = FALSE,
  initial_keys = c(),
  mustWork = FALSE,
  ...
)
```

## Arguments

|             |                                                                                                                              |
|-------------|------------------------------------------------------------------------------------------------------------------------------|
| dir         | character vector of directories to serve                                                                                     |
| addr        | address                                                                                                                      |
| prefix      | character vector of server prefixes (must have same length as dir)                                                           |
| blocking    | logical, if FALSE runs in background and returns a handle                                                                    |
| cors        | logical, enable CORS headers                                                                                                 |
| coop        | logical, enable COOP/COEP headers                                                                                            |
| tls         | Logical to enable TLS (HTTPS), or an object returned by createTLSCertificate() to enable TLS with generated PEM credentials. |
| certfile    | path to TLS certificate file                                                                                                 |
| keyfile     | path to TLS key file                                                                                                         |
| silent      | logical, suppress server logs                                                                                                |
| log_handler | function, custom log handler function(handler, message, user)                                                                |
| auth_keys   | character vector of API keys for authentication. Default c() = no auth                                                       |

|                           |                                                                                                                                                  |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>auth</code>         | logical, enable dynamic authentication system (non-blocking mode only)                                                                           |
| <code>initial_keys</code> | character vector of initial API keys for dynamic auth system                                                                                     |
| <code>mustWork</code>     | logical, if TRUE and non-blocking, will check if server actually started and throw error if it failed (default FALSE for backward compatibility) |
| <code>...</code>          | additional arguments passed to the server                                                                                                        |

## Value

NULL (if blocking) or an external pointer (if non-blocking)

## Examples

```
## Not run:
# Start a blocking server (will block the R session)
# runServer(dir = ".", addr = "0.0.0.0:8080")

# Start a background server (returns a handle)
h <- runServer(dir = ".", addr = "0.0.0.0:8080", blocking = FALSE)

# Start a server with static auth keys (backward compatible)
h <- runServer(
  dir = ".", addr = "0.0.0.0:8080", blocking = FALSE,
  auth_keys = c("secret123", "token456")
)

# Start a server with dynamic auth system
h <- runServer(
  dir = ".", addr = "0.0.0.0:8080", blocking = FALSE,
  auth = TRUE, initial_keys = c("secret123")
)

# Manage auth keys dynamically (only with auth=TRUE)
auth <- attr(h, "auth")
addAuthKey(auth, "new_key_456")
removeAuthKey(auth, "secret123")
listAuthKeys(auth)

# Start a server serving multiple directories
h <- runServer(
  dir = c("./data", "./docs", "."),
  prefix = c("/api/data", "/docs", "/files"),
  addr = "0.0.0.0:8080",
  blocking = FALSE
)

# Start a server with custom log handler
logfile <- tempfile("server_", fileext = ".log")
h <- runServer(
  dir = ".", addr = "0.0.0.0:8080", blocking = FALSE,
  log_handler = function(handler, message, user) {
    cat("[CUSTOM]", message, file = logfile, append = TRUE)
  }
)
```

```

    }
)

# List all running background servers
listServers()

# Get a summary view
summary(listServers())

# Shutdown a background server
shutdownServer(h)

## End(Not run)

```

---

|                |                                                    |
|----------------|----------------------------------------------------|
| shutdownServer | <i>shutdownServer Shutdown a background server</i> |
|----------------|----------------------------------------------------|

---

**Description**

shutdownServer Shutdown a background server

**Usage**

```
shutdownServer(handle)
```

**Arguments**

|        |                                                        |
|--------|--------------------------------------------------------|
| handle | external pointer returned by runServer(blocking=FALSE) |
|--------|--------------------------------------------------------|

---

|             |                                                                             |
|-------------|-----------------------------------------------------------------------------|
| StartServer | <i>StartServer (advanced/manual use) Start a server (C-level, advanced)</i> |
|-------------|-----------------------------------------------------------------------------|

---

**Description**

StartServer (advanced/manual use) Start a server (C-level, advanced)

**Usage**

```

StartServer(
  dir,
  addr,
  prefix,
  blocking,
  cors = FALSE,
  coop = FALSE,
  tls = FALSE,
  certfile = "cert.pem",

```

```

    keyfile = "key.pem",
    silent = FALSE,
    log_handler = NULL,
    auth_keys = c()
  )

```

### Arguments

|             |                                                                    |
|-------------|--------------------------------------------------------------------|
| dir         | character vector of directories to serve                           |
| addr        | address                                                            |
| prefix      | character vector of server prefixes (must have same length as dir) |
| blocking    | logical, if FALSE runs in background and returns a handle          |
| cors        | logical, enable CORS headers                                       |
| coop        | logical, enable COOP/COEP headers                                  |
| tls         | logical, enable TLS (HTTPS)                                        |
| certfile    | path to TLS certificate file                                       |
| keyfile     | path to TLS key file                                               |
| silent      | logical, suppress server logs                                      |
| log_handler | function, custom log handler function(handler, message, user)      |
| auth_keys   | character vector of API keys for authentication                    |

---

|                     |                                               |
|---------------------|-----------------------------------------------|
| summary.server_list | <i>Summary method for server_list objects</i> |
|---------------------|-----------------------------------------------|

---

### Description

Summary method for server\_list objects

### Usage

```

## S3 method for class 'server_list'
summary(object, ...)

```

### Arguments

|        |                                |
|--------|--------------------------------|
| object | a server_list object           |
| ...    | additional arguments (ignored) |

---

|               |                                     |
|---------------|-------------------------------------|
| updateAuthKey | <i>Update an Authentication Key</i> |
|---------------|-------------------------------------|

---

**Description**

Replaces an existing API key while keeping the replacement valid throughout the operation by adding the new key before removing the old key.

**Usage**

```
updateAuthKey(server_handle, old_key, new_key)
```

**Arguments**

|               |                                                                 |
|---------------|-----------------------------------------------------------------|
| server_handle | External pointer from runServer(blocking = FALSE, auth = TRUE). |
| old_key       | Existing API key.                                               |
| new_key       | Replacement API key.                                            |

**Value**

Invisible TRUE.

# Index

`.create_default_log_handler`, [2](#)  
`.default_log_callback`, [3](#)  
  
`addAuthKey`, [3](#)  
`as.data.frame.server_list`, [4](#)  
  
`clearAuthKeys`, [4](#)  
`createFileLogHandler`, [5](#)  
`createSilentLogHandler`, [5](#)  
`createTLSCertificate`, [6](#)  
  
`isRunning`, [6](#)  
  
`listAuthKeys`, [7](#)  
`listServers`, [7](#)  
  
`print.server_info`, [8](#)  
`print.server_list`, [8](#)  
  
`registerLogHandler`, [9](#)  
`removeAuthKey`, [9](#)  
`removeLogHandler`, [10](#)  
`removeTLSCertificate`, [10](#)  
`runServer`, [11](#)  
  
`shutdownServer`, [13](#)  
`StartServer`, [13](#)  
`summary.server_list`, [14](#)  
  
`updateAuthKey`, [15](#)