# Package 'needenv'

August 5, 2026

**Title** Validate Required Environment Variables with Defaults

**Version** 0.1.0

**Description** Validates that required environment variables are set and non-empty before environment-dependent code runs. Variables may have call-site defaults, and all unresolved variables are reported together. Successful values are returned as a named list without reading environment files or modifying the process environment.

**URL** https://github.com/cole-brokamp/needenv

**BugReports** https://github.com/cole-brokamp/needenv/issues

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 8.0.0

**Depends** R (>= 4.0.0)

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Cole Brokamp [aut, cre]

**Maintainer** Cole Brokamp <cole@colebrokamp.com>

**Repository** CRAN

**Date/Publication** 2026-08-05 09:20:18 UTC

## Contents

needenv *Validate required environment variables*

## Description

needenv() checks that a set of environment variables is available and non-empty. Variables may be supplied with defaults. All unresolved variables are reported together, and all variables that use defaults are reported in one warning.

## Usage

```
needenv(..., .vars = NULL)
```

## Arguments

| | |
|---|---|
| ... | Bare or quoted environment-variable names. Unnamed arguments are required. Named arguments associate the argument name with a default value. Bare names are preferred. Do not supply ... together with .vars. |
| .vars | NULL, or a character vector describing variables programmatically. Unnamed elements contain required variable names; named elements associate their names with defaults. Do not supply .vars together with .... |

## Details

Each variable is resolved independently. A set, non-empty process environment value is used first. Otherwise, a supplied default is used. A variable without either value is missing. Defaults are returned to the caller but are never written to the process environment. Named default expressions in ... are evaluated only when their environment value is unavailable.

needenv() only inspects the current process environment. It does not read environment files and does not call Sys.setenv(). Package authors should call it at the point where configuration is needed rather than while their package is loading.

## Value

Invisibly, a needenv_config object: a named list of scalar character values whose print method redacts all values. If any variables use defaults, a warning of class needenv_default is signaled first. If any variables are unresolved, an error of class needenv_missing is signaled and no value is returned.

## R startup environment files

R reads site and user environment files before evaluating startup profiles. A project or user .Renviron file may therefore prepare variables before needenv() runs. R_ENVIRON can be set before R starts to select a site environment file, and R_ENVIRON_USER can select a user environment file. For example, a shell can launch a script with R_ENVIRON=.env Rscript analysis.R. These files must use R's Renviron syntax, and setting the selector inside an already-running R process is too late for startup processing. See Startup.

## Examples

```
variable_names <- c("NEEDENV_EXAMPLE_TOKEN", "NEEDENV_EXAMPLE_URL")
old_values <- Sys.getenv(variable_names, unset = NA_character_, names = TRUE)

Sys.setenv(
  NEEDENV_EXAMPLE_TOKEN = "example-token",
  NEEDENV_EXAMPLE_URL = "https://configured.example.com"
)

config <- needenv(
  NEEDENV_EXAMPLE_TOKEN,
  NEEDENV_EXAMPLE_URL = "https://default.example.com"
)

# printing redacts values
config

# but values are still accessible
config$NEEDENV_EXAMPLE_TOKEN
config$NEEDENV_EXAMPLE_URL

spec <- c(
  "NEEDENV_EXAMPLE_TOKEN",
  NEEDENV_EXAMPLE_URL = "https://default.example.com"
)
needenv(.vars = spec)

Sys.unsetenv(variable_names)
restore <- !is.na(old_values)
if (any(restore)) {
  do.call(Sys.setenv, as.list(old_values[restore]))
}
```

# Index