

Package: using (via r-universe)

July 16, 2026

Type Package

Title Add version constraints to library() calls

Version 0.4.0

Description Add version constraints and remote git repositories to library() calls. In interactive mode the user is prompted to install.

License MIT + file LICENSE

URL <https://github.com/anthonymnorth/using>

BugReports <https://github.com/anthonymnorth/using/issues>

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.0

Depends R (>= 3.3.0)

Imports utils, uuid, remotes, knitr, stats, withr

Suggests testthat (>= 2.1.0)

Config/pak/sysreqs git

Repository <https://milesmbain.r-universe.dev>

Date/Publication 2025-09-01 04:26:13 UTC

RemoteUrl <https://github.com/anthonymnorth/using>

RemoteRef master

RemoteSha ce032ebb0f7922b900450e1c9f34be13a66f069c

Contents

detect_dependencies	2
pkg	3
using	4

Index	5
-------	---

detect_dependencies	<i>detect_dependencies</i>
---------------------	----------------------------

Description

detect usage of using::pkg in a source file

Usage

```
detect_dependencies(file_path)
```

Arguments

file_path a length 1 character vector file path to an .R or .Rmd file

Details

Intended for use when building lock files (e.g. for renv) from source files.

returns a data.frame of all information in using::pkg calls with columns: package, min_version, repo. The value of a row may be NA if the argument was not supplied.

Each result row is guaranteed to be unique, however the result data frame may contain near duplicates, e.g. using::pkg(janitor, min_version = "2.0.1") and using::pkg(janitor, min_version = "1.0.0") would each create a row in the result data frame if they appeared in the same file, due to differing min_version.

Each using::pkg call is parsed based on it's literal expression and no variable substitution is done. So using(janitor, min_version = janitor_ver) would place "janitor_ver" in the min_row column of the result for this dependency.

Value

a data.frame summarising found using::pkg calls in the supplied file.

Author(s)

Miles McBain

pkg	<i>using::pkg</i>
-----	-------------------

Description

Optionally add version constraints to library calls

Usage

```
pkg(package, min_version = NULL, repo = NULL, ...)
```

Arguments

package	{character name} package name
min_version	{package_version character} min package version
repo	{character} repo url
...	additional arguments to pass to base::library()

Details

This function is intended to always be called with the `using::` prefix. A call without this prefix will throw a warning. This simplifies detecting dependencies declared by `using::pkg` for use with `{renv}` and `{capsule}`

Value

{character | logical} attached packages, or boolean if `logical.return` is TRUE

Examples

```
## Not run:
using::pkg(capsule)
using::pkg(dplyr, min_version = "0.8.0")
using::pkg(h3jsr, min_version = "1.0.0", repo = "https://github.com/obrl-soil/h3jsr")

## End(Not run)
```

`using`*using*

Description

Optionally add version constraints to library calls

Usage

```
using(...)
```

Arguments

... arguments to pass to `using::pkg()`

Details

This function is deprecated. See `using::pkg()`.

Value

{character | logical} attached packages, or boolean if `logical.return` is TRUE

Examples

```
## Not run:
using::pkg(capsule)
using::pkg(dplyr, min_version = "0.8.0")
using::pkg(h3jsr, min_version = "1.0.0", repo = "https://github.com/obrl-soil/h3jsr")

## End(Not run)
```

Index

`base::library()`, [3](#)

`detect_dependencies`, [2](#)

`pkg`, [3](#)

`using`, [4](#)

`using::pkg()`, [4](#)