

Package ‘rJavaEnv’

April 4, 2025

Title 'Java' Environments for R Projects

Version 0.3.0

Description Quickly install 'Java Development Kit (JDK)' without administrative privileges and set environment variables in current R session or project to solve common issues with 'Java' environment management in 'R'. Recommended to users of 'Java'/'rJava'-dependent 'R' packages such as 'r5r', 'opentripplanner', 'xlsx', 'openNLP', 'rWeka', 'RJDBC', 'tabulapdf', and many more. 'rJavaEnv' prevents common problems like 'Java' not found, 'Java' version conflicts, missing 'Java' installations, and the inability to install 'Java' due to lack of administrative privileges. 'rJavaEnv' automates the download, installation, and setup of the 'Java' on a per-project basis by setting the relevant 'JAVA_HOME' in the current 'R' session or the current working directory (via '.Rprofile', with the user's consent). Similar to what 'renv' does for 'R' packages, 'rJavaEnv' allows different 'Java' versions to be used across different projects, but can also be configured to allow multiple versions within the same project (e.g. with the help of 'targets' package). Note: there are a few extra steps for 'Linux' users, who don't have any 'Java' previously installed in their system, and who prefer package installation from source, rather than installing binaries from 'Posit Package Manager'. See documentation for details.

License MIT + file LICENSE

URL <https://github.com/e-kotov/rJavaEnv>,
<https://www.ekotov.pro/rJavaEnv/>

BugReports <https://github.com/e-kotov/rJavaEnv/issues>

Depends R (>= 4.0)

Imports checkmate, cli, curl, jsonlite, utils

Suggests quarto, rJava, testthat (>= 3.0.0)

VignetteBuilder quarto

Config/testthat/edition 3

Encoding UTF-8

Language en-US**RoxygenNote** 7.3.2**NeedsCompilation** no

Author Egor Kotov [aut, cre, cph] (<<https://orcid.org/0000-0001-6690-5345>>),
 Chung-hong Chan [aut] (<<https://orcid.org/0000-0002-6232-7530>>),
 Mauricio Vargas [ctb] (<<https://orcid.org/0000-0003-1017-7574>>),
 Hadley Wickham [ctb] (use_java feature suggestion and PR review),
 Enrique Mondragon-Estrada [ctb]
 (<<https://orcid.org/0009-0004-5592-1728>>),
 Jonas Lieth [ctb] (<<https://orcid.org/0000-0002-3451-3176>>)

Maintainer Egor Kotov <kotov.egor@gmail.com>**Repository** CRAN**Date/Publication** 2025-04-04 13:20:02 UTC

Contents

java_check_version_cmd	2
java_check_version_rjava	3
java_clear	4
java_download	5
java_env_set	6
java_env_unset	7
java_install	8
java_list	9
java_quick_install	10
java_unpack	11
java_valid_versions	12
rje_consent	13
use_java	14
Index	16

 java_check_version_cmd

Check installed Java version using terminal commands

Description

Check installed Java version using terminal commands

Usage

```
java_check_version_cmd(java_home = NULL, quiet = FALSE)
```

Arguments

java_home	Path to Java home directory. If NULL, the function uses the JAVA_HOME environment variable.
quiet	A logical value indicating whether to suppress messages. Can be TRUE or FALSE.

Value

A character vector of length 1 containing the major Java version.

Examples

```
java_check_version_cmd()
```

```
java_check_version_rjava
```

Check Java Version with a Specified JAVA_HOME Using a Separate R Session

Description

This function sets the JAVA_HOME environment variable, initializes the JVM using rJava, and prints the Java version that would be used if the user sets the given JAVA_HOME in the current R session. This check is performed in a separate R session to avoid having to reload the current R session. The reason for this is that once Java is initialized in an R session, it cannot be uninitialized unless the current R session is restarted.

Usage

```
java_check_version_rjava(java_home = NULL, quiet = FALSE)
```

Arguments

java_home	Path to Java home directory. If NULL, the function uses the JAVA_HOME environment variable.
quiet	A logical value indicating whether to suppress messages. Can be TRUE or FALSE.

Value

A character vector of length 1 containing the major Java version.

Examples

```
## Not run:
java_check_version_rjava()

## End(Not run)
```

 java_clear

Manage Java installations and distributions caches

Description

Wrapper function to clear the Java symlinked in the current project, installed, or distributions caches.

Usage

```
java_clear(
  type = c("project", "installed", "distrib"),
  target_dir = NULL,
  check = TRUE,
  delete_all = FALSE
)
```

Arguments

type	What to clear: "project" - remove symlinks to install cache in the current project, "installed" - remove installed Java versions, "distrib" - remove downloaded Java distributions.
target_dir	The directory to clear. Defaults to current working directory for "project" and user-specific data directory for "installed" and "distrib". Not recommended to change.
check	Whether to list the contents of the cache directory before clearing it. Defaults to TRUE.
delete_all	Whether to delete all items without prompting. Defaults to FALSE.

Value

A message indicating whether the cache was cleared or not.

Examples

```
## Not run:
java_clear("project", target_dir = tempdir())
java_clear("installed", target_dir = tempdir())
java_clear("distrib", target_dir = tempdir())

## End(Not run)
```

java_download	<i>Download a Java distribution</i>
---------------	-------------------------------------

Description

Download a Java distribution

Usage

```
java_download(
  version = 21,
  distribution = "Corretto",
  cache_path = getOption("rJavaEnv.cache_path"),
  platform = platform_detect()$os,
  arch = platform_detect()$arch,
  quiet = FALSE,
  force = FALSE,
  temp_dir = FALSE
)
```

Arguments

version	Integer or character vector of length 1 for major version of Java to download or install. If not specified, defaults to the latest LTS version. Can be "8", and "11" to "24" (or the same version numbers in integer) or any newer version if it is available for the selected distribution. For macOS on aarch64 architecture (Apple Silicon) certain Java versions are not available.
distribution	The Java distribution to download. If not specified, defaults to "Amazon Corretto". Currently only "Amazon Corretto" is supported.
cache_path	The destination directory to download the Java distribution to. Defaults to a user-specific data directory.
platform	The platform for which to download the Java distribution. Defaults to the current platform.
arch	The architecture for which to download the Java distribution. Defaults to the current architecture.
quiet	A logical value indicating whether to suppress messages. Can be TRUE or FALSE.
force	A logical. Whether the distribution file should be overwritten or not. Defaults to FALSE.
temp_dir	A logical. Whether the file should be saved in a temporary directory. Defaults to FALSE.

Value

The path to the downloaded Java distribution file.

Examples

```
## Not run:

# download distribution of Java version 17
java_download(version = "17", temp_dir = TRUE)

# download default Java distribution (version 21)
java_download(temp_dir = TRUE)

## End(Not run)
```

java_env_set	<i>Set the JAVA_HOME and PATH environment variables to a given path</i>
--------------	---

Description

Set the JAVA_HOME and PATH environment variables to a given path

Usage

```
java_env_set(
  where = c("session", "both", "project"),
  java_home,
  project_path = NULL,
  quiet = FALSE
)
```

Arguments

where	Where to set the JAVA_HOME: "session", "project", or "both". Defaults to "session" and only updates the paths in the current R session. When "both" or "project" is selected, the function updates the .Rprofile file in the project directory to set the JAVA_HOME and PATH environment variables at the start of the R session.
java_home	The path to the desired JAVA_HOME.
project_path	A character vector of length 1 containing the project directory where Java should be installed. If not specified or NULL, defaults to the current working directory.
quiet	A logical value indicating whether to suppress messages. Can be TRUE or FALSE.

Value

Nothing. Sets the JAVA_HOME and PATH environment variables.

Examples

```
## Not run:
# download, install Java 17
java_17_distrib <- java_download(version = "17", temp_dir = TRUE)
java_home <- java_install(
  java_distrib_path = java_17_distrib,
  project_path = tempdir(),
  autosest_java_env = FALSE
)

# now manually set the JAVA_HOME and PATH environment variables in current session
java_env_set(
  where = "session",
  java_home = java_home
)

# or set JAVA_HOME and PATH in the specific projects' .Rprofile
java_env_set(
  where = "session",
  java_home = java_home,
  project_path = tempdir()
)

## End(Not run)
```

java_env_unset	<i>Unset the JAVA_HOME and PATH environment variables in the project .Rprofile</i>
----------------	--

Description

Unset the JAVA_HOME and PATH environment variables in the project .Rprofile

Usage

```
java_env_unset(project_path = NULL, quiet = FALSE)
```

Arguments

project_path	A character vector of length 1 containing the project directory where Java should be installed. If not specified or NULL, defaults to the current working directory.
quiet	A logical value indicating whether to suppress messages. Can be TRUE or FALSE.

Value

Nothing. Removes the JAVA_HOME and PATH environment variables settings from the project .Rprofile.

Examples

```
## Not run:  
# clear the JAVA_HOME and PATH environment variables in the specified project .Rprofile  
java_env_unset(project_path = tempdir())  
  
## End(Not run)
```

java_install

Install Java from a distribution file

Description

Unpack Java distribution file into cache directory and link the installation into a project directory, optionally setting the JAVA_HOME and PATH environment variables to the Java version that was just installed.

Usage

```
java_install(  
  java_distrib_path,  
  project_path = NULL,  
  autosest_java_env = TRUE,  
  quiet = FALSE  
)
```

Arguments

java_distrib_path	A character vector of length 1 containing the path to the Java distribution file.
project_path	A character vector of length 1 containing the project directory where Java should be installed. If not specified or NULL, defaults to the current working directory.
autosest_java_env	A logical indicating whether to set the JAVA_HOME and PATH environment variables to the installed Java directory. Defaults to TRUE.
quiet	A logical value indicating whether to suppress messages. Can be TRUE or FALSE.

Value

The path to the installed Java directory.

Examples

```
## Not run:

# set cache dir to temporary directory
options(rJavaEnv.cache_path = tempdir())
# download, install and autoset environment variables for Java 17
java_17_distrib <- java_download(version = "17")
java_install(java_distrib_path = java_17_distrib, project_path = tempdir())

## End(Not run)
```

java_list

List the contents of the Java versions installed or cached

Description

This function lists one of the following:

- `project` - list the contents of the Java symlinked/copied in the current project or directory specified by `target_dir`
- `distrib` - list the contents of the downloaded Java distributions cache in default location or specified by `target_dir`
- `installed` - list the contents of the Java installations cache (unpacked distributions) in default location or specified by `target_dir`

Usage

```
java_list(
  type = c("project", "installed", "distrib"),
  output = c("data.frame", "vector"),
  quiet = TRUE,
  target_dir = NULL
)
```

Arguments

<code>type</code>	The type of cache to list: "distrib", "installed", or "project". Defaults to "project".
<code>output</code>	The format of the output: "data.frame" or "vector". Defaults to "data.frame".
<code>quiet</code>	A logical value indicating whether to suppress messages. Can be TRUE or FALSE.
<code>target_dir</code>	The cache directory to list. Defaults to the user-specific data directory for "distrib" and "installed", and the current working directory for "project".

Value

A dataframe or character vector with the contents of the specified cache or project directory.

Examples

```
## Not run:
java_list("project")
java_list("installed")
java_list("distrib")

## End(Not run)
```

java_quick_install	<i>Download and install and set Java in current working/project directory</i>
--------------------	---

Description

Download and install and set Java in current working/project directory

Usage

```
java_quick_install(
  version = 21,
  distribution = "Corretto",
  project_path = NULL,
  platform = platform_detect()$os,
  arch = platform_detect()$arch,
  quiet = FALSE,
  temp_dir = FALSE
)
```

Arguments

version	Integer or character vector of length 1 for major version of Java to download or install. If not specified, defaults to the latest LTS version. Can be "8", and "11" to "24" (or the same version numbers in integer) or any newer version if it is available for the selected distribution. For macOS on aarch64 architecture (Apple Silicon) certain Java versions are not available.
distribution	The Java distribution to download. If not specified, defaults to "Amazon Corretto". Currently only "Amazon Corretto" is supported.
project_path	A character vector of length 1 containing the project directory where Java should be installed. If not specified or NULL, defaults to the current working directory.
platform	The platform for which to download the Java distribution. Defaults to the current platform.
arch	The architecture for which to download the Java distribution. Defaults to the current architecture.

quiet	A logical value indicating whether to suppress messages. Can be TRUE or FALSE.
temp_dir	A logical. Whether the file should be saved in a temporary directory. Defaults to FALSE.

Value

Message indicating that Java was installed and set in the current working/project directory.

Examples

```
## Not run:

# quick download, unpack, install and set in current working directory default Java version (21)
java_quick_install(17, temp_dir = TRUE)

## End(Not run)
```

java_unpack	<i>Unpack a Java distribution file into cache directory</i>
-------------	---

Description

Unpack the Java distribution file into cache directory and return the path to the unpacked Java directory with Java binaries.

Usage

```
java_unpack(java_distrib_path, quiet = FALSE)
```

Arguments

java_distrib_path	A character vector of length 1 containing the path to the Java distribution file.
quiet	A logical value indicating whether to suppress messages. Can be TRUE or FALSE.

Value

A character vector containing of length 1 containing the path to the unpacked Java directory.

Examples

```
## Not run:

# set cache dir to temporary directory
options(rJavaEnv.cache_path = tempdir())

# download Java 17 distrib and unpack it into cache dir
java_17_distrib <- java_download(version = "17")
java_home <- java_unpack(java_distrib_path = java_17_distrib)

# set the JAVA_HOME environment variable in the current session
# to the cache dir without touching any files in the current project directory
java_env_set(where = "session", java_home = java_home)

## End(Not run)
```

java_valid_versions *Retrieve Valid Java Versions*

Description

This function retrieves a list of valid Java versions by querying an appropriate API endpoint based on the chosen distribution. The result is cached for 8 hours to avoid repeated API calls. If the API call fails (for example, due to a lack of internet connectivity), the function falls back to a pre-defined list of Java versions.

Usage

```
java_valid_versions(  
  distribution = "Corretto",  
  platform = platform_detect()$os,  
  arch = platform_detect()$arch,  
  force = FALSE  
)
```

Arguments

distribution	The Java distribution to download. If not specified, defaults to "Amazon Corretto". Currently only " Amazon Corretto " is supported.
platform	The platform for which to download the Java distribution. Defaults to the current platform.
arch	The architecture for which to download the Java distribution. Defaults to the current architecture.
force	Logical. If TRUE, forces a fresh API call even if a cached value exists. Defaults to FALSE.

Value

A character vector of valid Java versions.

Examples

```
## Not run:
# Retrieve valid Java versions (cached if available) using Amazon Corretto endpoint
versions <- java_valid_versions()

# Force refresh the list of Java versions using the Oracle endpoint
versions <- java_valid_versions(distribution = "Corretto", force = TRUE)

## End(Not run)
```

rje_consent

Obtain User Consent for rJavaEnv

Description

Get user consent for rJavaEnv to write and update files on the file system. rJavaEnv needs permission to manage files in your project and cache directories to function correctly.

Usage

```
rje_consent(provided = FALSE)
```

Arguments

provided	Logical indicating if consent is already provided. To provide consent in non-interactive R sessions use <code>rJavaEnv::rje_consent(provided = TRUE)</code> . Default is FALSE.
----------	---

Details

In line with **CRAN policies**, explicit user consent is required before making these changes. Please call `rJavaEnv::consent()` to provide consent.

Alternatively, you can set the following R option (especially useful for non-interactive R sessions):

```
options(rJavaEnv.consent = TRUE)
```

The function is based on the code of the `renv` package. Copyright 2023 Posit Software, PBC
License: <https://github.com/rstudio/renv/blob/main/LICENSE>

Value

TRUE if consent is given, otherwise an error is raised.

Examples

```
## Not run:

# to provide consent and prevent other functions from interrupting to get the consent
rje_consent(provided = TRUE)

## End(Not run)
```

use_java	<i>Install specified Java version and set the JAVA_HOME and PATH environment variables in current R session</i>
----------	---

Description

Using specified Java version, set the JAVA_HOME and PATH environment variables in the current R session. If Java distribution has not been downloaded yet, download it. If it was not installed into cache directory yet, install it there and then set the environment variables. This is intended as a quick and easy way to use different Java versions in R scripts that are in the same project, but require different Java versions. For example, one could use this in scripts that are called by targets package or callr package.

Usage

```
use_java(
  version = NULL,
  distribution = "Corretto",
  cache_path = getOption("rJavaEnv.cache_path"),
  platform = platform_detect()$os,
  arch = platform_detect()$arch,
  quiet = TRUE
)
```

Arguments

version	Integer or character vector of length 1 for major version of Java to download or install. If not specified, defaults to the latest LTS version. Can be "8", and "11" to "24" (or the same version numbers in integer) or any newer version if it is available for the selected distribution. For macOS on aarch64 architecture (Apple Silicon) certain Java versions are not available.
distribution	The Java distribution to download. If not specified, defaults to "Amazon Corretto". Currently only "Amazon Corretto" is supported.
cache_path	The destination directory to download the Java distribution to. Defaults to a user-specific data directory.
platform	The platform for which to download the Java distribution. Defaults to the current platform.

arch	The architecture for which to download the Java distribution. Defaults to the current architecture.
quiet	A logical value indicating whether to suppress messages. Can be TRUE or FALSE.

Value

NULL. Prints the message that Java was set in the current R session if quiet is set to FALSE.

Examples

```
## Not run:

# set cache directory for Java to be in temporary directory
options(rJavaEnv.cache_path = tempdir())

# install and set Java 8 in current R session
use_java(8)
# check Java version
"8" == java_check_version_cmd(quiet = TRUE)
"8" == java_check_version_rjava(quiet = TRUE)

# install and set Java 17 in current R session
use_java(17)
# check Java version
"17" == java_check_version_cmd(quiet = TRUE)
"17" == java_check_version_rjava(quiet = TRUE)

## End(Not run)
```

Index

java_check_version_cmd, 2
java_check_version_rjava, 3
java_clear, 4
java_download, 5
java_env_set, 6
java_env_unset, 7
java_install, 8
java_list, 9
java_quick_install, 10
java_unpack, 11
java_valid_versions, 12

rje_consent, 13

use_java, 14