Package ‘rocrateR’

November 7, 2025

Title RO-Crate R Package Wrapper

Version 0.0.1

Description R package for creating, manipulating and reading RO-Crates. Latest supported version of the specification: <https://w3id.org/ro/crate/1.2/>.

License MIT + file LICENSE

Suggests spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

URL https://github.com/ResearchObject/ro-crate-r/

BugReports https://github.com/ResearchObject/ro-crate-r/issues/

RoxygenNote 7.3.3

Imports digest, jsonlite, tibble, zip

Depends R (>= 4.1.0)

NeedsCompilation no

Author Roberto Villegas-Diaz [aut, cre] (ORCID: <https://orcid.org/0000-0001-5036-8661>),
Research Object community [cph]

Maintainer Roberto Villegas-Diaz <r.villegas-diaz@outlook.com>

Repository CRAN

Date/Publication 2025-11-07 14:00:09 UTC

Contents

`add_entities` *Wrapper for add_entity*

Description

Wrapper for add_entity, can be use to add multiple entities.

Usage

```
add_entities(rocrate, entity, overwrite = FALSE, quiet = FALSE)
```

Arguments

`rocrate`	RO-Crate object, see rocrate.
`entity`	List with entity objects.
`overwrite`	Boolean flag to indicate if the entity (if found in the given RO-Crate) should be overwritten.
`quiet`	Boolean flag to indicate if status messages should be hidden (default: `FALSE`).

Value

Updated RO-Crate with the new entities.

`add_entity` *Add entity to RO-Crate*

Description

Add entity to RO-Crate

Usage

```
add_entity(rocrate, entity, overwrite = FALSE)
```

Arguments

`rocrate` RO-Crate object, see rocrate.

`entity` Entity object (list) that contains at least the following components: `@id` and `@type`.

`overwrite` Boolean flag to indicate if the entity (if found in the given RO-Crate) should be overwritten.

Value

Updated RO-Crate with the new entity.

Examples

```
basic_crate <- rocrateR::rocrate()

# create entity for an organisation
organisation_uol <- rocrateR::entity(
  x = "https://ror.org/04xs57h96",
  type = "Organization",
  name = "University of Liverpool",
  url = "http://www.liv.ac.uk"
)

# create an entity for a person
person_rvd <- rocrateR::entity(
  x = "https://orcid.org/0000-0001-5036-8661",
  type = "Person",
  name = "Roberto Villegas-Diaz",
  affiliation = list(`@id` = organisation_uol$`@id`)
)

basic_crate_v2 <- basic_crate |>
  rocrateR::add_entity(person_rvd) |>
  rocrateR::add_entity_value(id = "./", key = "author", value = list(`@id` = person_rvd$`@id`)) |>
  rocrateR::add_entity(organisation_uol)
```

add_entity_value *Add entity value to RO-Crate*

Description

Add entity value to RO-Crate, under entity with `@id` = `{id}`, using the pair `{key}`-`{value}` within `@graph`.

Usage

```
add_entity_value(rocrate, id, key, value, overwrite = TRUE)
```

Arguments

`rocrate`	RO-Crate object, see rocrate.
`id`	String with the ID of the RO-Crate entity within `@graph`.
`key`	String with the `key` of the entity with `id` to be modified.
`value`	String with the `value` for `key`.
`overwrite`	Boolean flag to indicate if the existing value (if any), should be overwritten (default: `TRUE`).

Value

RO-Crate object.

Examples

```
basic_crate <- rocrate()

# create entity for an organisation
organisation_uol <- rocrateR::entity(
  x = "https://ror.org/04xs57h96",
  type = "Organization",
  name = "University of Liverpool",
  url = "http://www.liv.ac.uk"
)

# create an entity for a person
person_rvd <- rocrateR::entity(
  x = "https://orcid.org/0000-0001-5036-8661",
  type = "Person",
  name = "Roberto Villegas-Diaz",
  affiliation = list(`@id` = organisation_uol$`@id`)
)

basic_crate_v2 <- basic_crate |>
 rocrateR::add_entity_value(id = "./", key = "author", value = list(`@id` = person_rvd$`@id`))
```

bag_rocrate *Bag the contents of an RO-Crate*

Description

Bag the contents of an RO-Crate using the BagIt file packaging format v1.0. For more details see the definition: doi:10.17487/RFC8493

Usage

```
bag_rocrate(x, ...)

## S3 method for class 'character'
bag_rocrate(x, ..., output = x, force_bag = FALSE)

## S3 method for class 'rocrate'
bag_rocrate(x, ..., path, output = path, overwrite = FALSE, force_bag = FALSE)
```

Arguments

`x`	A string to a path containing at the very minimum an RO-Crate metadata descriptor file, `ro-crate-metadata.json`. Alternatively, an object with the rocrate class.
`...`	Additional parameters, see below.
`output`	String with path where the RO-Crate bag will be stored (default: `x` - same path as the input value).
`force_bag`	Boolean flag to indicate whether the force the creation of a 'bag' even if not all the files were successfully bagged (default: `FALSE` ~ check if all the files were copied successfully).
`path`	String with path to the root of the RO-Crate.
`overwrite`	Boolean flag to indicate if the RO-Crate metadata descriptor file should be overwritten if already inside `path` (default: `FALSE`).

Value

String with full path to the final RO-Crate bag.

See Also

Other bag_rocrate: `is_rocrate_bag()`, `unbag_rocrate()`

`entity` *Create a data entity*

Description

Create a data entity

Usage

```
entity(x, ...)
```

Arguments

`x` New entity. If a single value (e.g., `character`, `numeric`) is given, this is assumed to be the entity's `@id`, if a `list` is given, this is assumed to be a complete entity. Other options are objects of type `person` and `organisation` (equivalently `organization`).

`...` Optional additional entity values/properties. Used when `x` is a single value.

Value

List with an entity object.

Examples

```
# create entity for an organisation
organisation_uol <- rocrateR::entity(
  x = "https://ror.org/04xs57h96",
  type = "Organization",
  name = "University of Liverpool",
  url = "http://www.liv.ac.uk"
)

# create an entity for a person
person_rvd <- rocrateR::entity(
  x = "https://orcid.org/0000-0001-5036-8661",
  type = "Person",
  name = "Roberto Villegas-Diaz",
  affiliation = list(`@id` = organisation_uol$`@id`)
)
```

get_entity *Get entity(ies)*

Description

Get entity(ies)

Usage

```
get_entity(rocrate, id = NULL, type = NULL)
```

Arguments

`rocrate`	RO-Crate object, see rocrate.
`id`	String with the ID of the RO-Crate entity within `@graph` (optional if `type` is provided). Alternatively, an entity object / list with `@id` and `@type`.
`type`	String with the type of the RO-Crate entity(ies) within `@graph` to retrieve (optional if `id` is provided).

Value

List with found entity object(s), if any, `NULL` otherwise.

Examples

```
basic_crate <- rocrateR::rocrate()

# create entity for an organisation
organisation_uol <- rocrateR::entity(
  x = "https://ror.org/04xs57h96",
  type = "Organization",
  name = "University of Liverpool",
  url = "http://www.liv.ac.uk"
)

# create an entity for a person
person_rvd <- rocrateR::entity(
  x = "https://orcid.org/0000-0001-5036-8661",
  type = "Person",
  name = "Roberto Villegas-Diaz",
  affiliation = list(`@id` = organisation_uol$`@id`)
)

basic_crate_person <- basic_crate |>
  rocrateR::add_entity(person_rvd) |>
 rocrateR::add_entity_value(id = "./", key = "author", value = list(`@id` = person_rvd$`@id`)) |>
  rocrateR::add_entity(organisation_uol) |>
  rocrateR::get_entity(person_rvd)
```

```
basic_crate_person[[1]]$name == person_rvd$name
basic_crate_person[[1]]$`@id` == person_rvd$`@id`
```

`is_rocrate` *Check if object is an RO-Crate*

Description

Check if object is an RO-Crate

Usage

```
is_rocrate(rocrate)
```

Arguments

`rocrate` RO-Crate object, see rocrate.

Value

Returns invisibly the input RO-Crate object.

Examples

```
basic_crate <- rocrateR::rocrate()

# check if the new crate is valid
basic_crate |>
  rocrateR::is_rocrate()
```

`is_rocrate_bag` *Check if path points to a valid RO-Crate bag*

Description

Check if path points to a valid RO-Crate bag

Usage

```
is_rocrate_bag(path, algo = "sha512", bagit_version = "1.0")
```

Arguments

`path` String with full path to a compressed file contain an RO-Crate bag, see bag_rocrate for details. Alternatively, a path to a directory containing an RO-Crate bag.

`algo` String with algorithm used to generate the RO-Crate bag (default: `"sha512"`). See digest for more details.

`bagit_version` String with version of BagIt used to generate the RO-Crate bag (default: `"1.0"`). See doi:10.17487/RFC8493 for more details.

Value

Returns invisibly the RO-Crate pointed by `path`.

See Also

Other bag_rocrate: `bag_rocrate()`, `unbag_rocrate()`

`print.entity` *Print RO-Crate entity*

Description

Print RO-Crate entity, S3 method for class 'entity'.

Usage

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

Arguments

`x`	RO-Crate entity object, see entity.
`...`	Optional arguments, not used.

Value

Invisibly the input RO-Crate entity, `x`.

Examples

```
rocrateR::rocrate() |>
  rocrateR::get_entity("./")
```

`print.rocrate` *Print RO-Crate*

Description

Print RO-Crate, S3 method for class 'rocrate'. Creates a temporal JSON file, which then is displayed with the message function.

Usage

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

Arguments

`x`	RO-Crate object, see rocrate.
`...`	Optional arguments, not used.

Value

Invisibly the input RO-Crate, `x`.

Examples

```
rocrateR::rocrate()
```

`read_rocrate` *Wrapper for jsonlite::read_json*

Description

Wrapper for jsonlite::read_json. Enforces that the object read is an RO-Crate.

Usage

```
read_rocrate(path, simplifyVector = FALSE, ...)
```

Arguments

`path` file on disk

`simplifyVector` simplifies nested lists into vectors and data frames. See `fromJSON()`.

`...` Arguments passed on to `jsonlite::fromJSON`

- `txt` a JSON string, URL or file
- `simplifyDataFrame` coerce JSON arrays containing only records (JSON objects) into a data frame
- `simplifyMatrix` coerce JSON arrays containing vectors of equal mode and dimension into matrix or array
- `flatten` automatically `flatten()` nested data frames into a single non-nested data frame

Value

Invisibly the RO-Crate stored in `path`.

`remove_entities` *Wrapper for remove_entity*

Description

Wrapper for remove_entity, can be use to remove multiple entities.

Usage

```
remove_entities(rocrate, entity)
```

Arguments

`rocrate`	RO-Crate object, see rocrate.
`entity`	Entity object (list) that contains at least the following components: `@id` and `@type`. Or, scalar value with entity `@id`.

Value

Updated RO-Crate.

`remove_entity` *Remove entity*

Description

Remove entity

Usage

```
remove_entity(rocrate, entity)
```

Arguments

`rocrate`	RO-Crate object, see rocrate.
`entity`	Entity object (list) that contains at least the following components: `@id` and `@type`. Or, scalar value with entity `@id`.

Value

Updated RO-Crate object.

Examples

```
basic_crate <- rocrateR::rocrate()

# create entity for an organisation
organisation_uol <- rocrateR::entity(
  x = "https://ror.org/04xs57h96",
  type = "Organization",
  name = "University of Liverpool",
  url = "http://www.liv.ac.uk"
)

# create an entity for a person
person_rvd <- rocrateR::entity(
  x = "https://orcid.org/0000-0001-5036-8661",
  type = "Person",
  name = "Roberto Villegas-Diaz",
  affiliation = list(`@id` = organisation_uol$`@id`)
)

basic_crate_v2 <- basic_crate |>
  rocrateR::add_entity(person_rvd) |>
 rocrateR::add_entity_value(id = "./", key = "author", value = list(`@id` = person_rvd$`@id`)) |>
  rocrateR::add_entity(organisation_uol) |>
  rocrateR::remove_entity(person_rvd)
```

---

rocrate *Create a new RO-Crate object*

Description

Create a new RO-Crate object. This object includes basic skeleton for the RO-Crate metadata descriptor (`ro-crate-metadata.json`) file, as described in the official documentation: https://w3id.org/ro/crate/1.2/ > Root Data Entity.

Usage

```
rocrate(
  ...,
  context = "https://w3id.org/ro/crate/1.2/context",
  conformsTo = gsub("\\/context$", "\\1", context),
  datePublished = Sys.Date(),
  description = "",
  license = "http://spdx.org/licenses/CC-BY-4.0",
  name = ""
)
```

Arguments

`...` Optional entities to include in the RO-Crate (e.g., author).

`context` String with URL to the version of the RO-Crate specification to use. The context brings the defined terms into the metadata document (default: https://w3id.org/ro/crate/1.2/context).

`conformsTo` String with URL to the version of the RO-Crate specification which this object conforms to. Conformance declares which RO-Crate conventions of using those terms are being followed (default: URL formed by `context`/context)

`datePublished` String (or Date object) with the date in which the RO-Crate was published (default: current date).

`description` String with description for the root entity (default: empty string).

`license` String with URL (permalinks are preferred, but not required) to license to be used for the overall RO-Crate. See the following resources for license choices: `https://spdx.org/licenses` and/or `https://github.com/spdx/license-list-data/tree/main/jsonld` (default: CC-BY-4.0: Creative Commons Attribution 4.0 International).

`name` String with a name/title for the root entity (default: empty string).

Value

RO-Crate object, list with an additional class, `rocrate`.

Examples

```
rocrateR::rocrate()
```

`rocrate_5s` *Create a new 5 Safes RO-Crate object*

Description

Create a new 5 Safes RO-Crate object. This object includes basic skeleton for the RO-Crate metadata descriptor (`ro-crate-metadata.json`) file, as described in the official documentation: https://w3id.org/ro/crate/1.2 > Root Data Entity. Additionally, it includes a profile for the 5 Safes RO-Crate: https://w3id.org/5s-crate/0.4

Usage

```
rocrate_5s(
  ...,
  context = "https://w3id.org/ro/crate/1.2/context",
  conformsTo = gsub("\\/context$", "\\1", context),
  datePublished = Sys.Date(),
  description = "",
  license = "http://spdx.org/licenses/CC-BY-4.0",
  name = "",
  v5scrate = 0.4
)
```

Arguments

`...` Optional entities to include in the RO-Crate (e.g., author).

`context` String with URL to the version of the RO-Crate specification to use. The context brings the defined terms into the metadata document (default: https://w3id.org/ro/crate/1.2/context).

`conformsTo` String with URL to the version of the RO-Crate specification which this object conforms to. Conformance declares which RO-Crate conventions of using those terms are being followed (default: URL formed by `context`/context)

`datePublished` String (or Date object) with the date in which the RO-Crate was published (default: current date).

`description` String with description for the root entity (default: empty string).

`license` String with URL (permalinks are preferred, but not required) to license to be used for the overall RO-Crate. See the following resources for license choices: `https://spdx.org/licenses` and/or `https://github.com/spdx/license-list-data/tree/main/jsonld` (default: CC-BY-4.0: Creative Commons Attribution 4.0 International).

`name` String with a name/title for the root entity (default: empty string).

`v5scrate` Numeric value with the version of the 5 Safes RO-Crate profile to use.

Value

5 Safes RO-Crate object, list with an additional class, `rocrate`.

Examples

```
rocrateR::rocrate_5s()
```

`unbag_rocrate` *'Unbag' (extract) RO-Crate packed with BagIt*

Description

'Unbag' (extract) RO-Crate packed with BagIt

Usage

```
unbag_rocrate(path, output = dirname(path), quiet = FALSE)
```

Arguments

`path` String with path to compressed file containing an RO-Crate bag.

`output` String with target path where the contents will be extracted (default: `dirname(path)` - same directory as input `path`).

`quiet` Boolean flag to indicate if messages should be suppressed (default: `FALSE` - display messages).

Value

String with path to root of the RO-Crate, invisibly.

See Also

Other bag_rocrate: `bag_rocrate()`, `is_rocrate_bag()`

`write_rocrate` *Wrapper for jsonlite::write_json*

Description

Wrapper for jsonlite::write_json. Enforces that the input object is an RO-Crate.

Usage

```
write_rocrate(x, path, ...)
```

Arguments

`x` RO-Crate object, see rocrate.

`path` file on disk

`...` Arguments passed on to `jsonlite::toJSON`

- `dataframe` how to encode data.frame objects: must be one of 'rows', 'columns' or 'values'
- `matrix` how to encode matrices and higher dimensional arrays: must be one of 'rowmajor' or 'columnmajor'.
- `Date` how to encode Date objects: must be one of 'ISO8601' or 'epoch'
- `POSIXt` how to encode POSIXt (datetime) objects: must be one of 'string', 'ISO8601', 'epoch' or 'mongo'
- `factor` how to encode factor objects: must be one of 'string' or 'integer'
- `complex` how to encode complex numbers: must be one of 'string' or 'list'
- `raw` how to encode raw objects: must be one of 'base64', 'hex' or 'mongo'
- `null` how to encode NULL values within a list: must be one of 'null' or 'list'
- `na` how to print NA values: must be one of 'null' or 'string'. Defaults are class specific
- `digits` max number of decimal digits to print for numeric values. Use `I()` to specify significant digits. Use `NA` for max precision.
- `force` unclass/skip objects of classes with no defined JSON mapping

Value

Invisibly the input RO-Crate, `x`.

Index