Package: eqty.sdk.r (via r-universe)

October 15, 2026

Title Verifiable Provenance for 'R' Workflows

Version 0.9.2

Description Provides an 'R' interface for building verifiable manifests around data, models, prompts, agents, computations, and related lineage. Exposes the public 'eqty-sdk' 'Python' package through types accessible from 'R' for creating assets, connecting computation inputs and outputs by content identifier, grouping statements in contexts, and exporting manifest JSON files.

License Apache License (== 2.0)

Encoding UTF-8

Imports reticulate (>= 1.41)

SystemRequirements Python (>= 3.10)

URL https://github.com/eqtylab/integrity-r, https://eqtylab.github.io/integrity-py/latest/, https://pypi.org/project/eqty-sdk/

BugReports https://github.com/eqtylab/integrity-r/issues

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Support EQTY Lab [aut, cre]

Maintainer Support EQTY Lab <package-management@eqtylab.io>

Config/pak/sysreqs python3

Repository https://cran.r-universe.dev

Date/Publication 2026-10-15 15:50:08 UTC

RemoteUrl https://github.com/cran/eqty.sdk.r

RemoteRef HEAD

RemoteSha fe0dd5afc7991abb4d328b0d181606953a74754c

Contents

compute . . . 2
eqty . . . 3
eqty-sdk-types . . . 3
get_cid_for_bytes . . . 4
get_cid_for_json . . . 4
get_cid_for_path . . . 5
integrity_compute . . . 5
integrity_dict . . . 7
integrity_init . . . 7
integrity_path . . . 8
integrity_resolve_path . . . 8
integrity_set_active_signer . . . 9
purge_blob_store . . . 9
purge_statement_store . . . 10
verify_statement . . . 10
verify_vc . . . 11

Index **12**

`compute` *Access the Python Compute Decorator*

Description

Provides direct access to ‘eqty_sdk.compute()‘. R functions should usually use [integrity_compute()], which executes the R function directly and records its source and execution with [Code] and [Computation].

Usage

```
compute(...)
```

Arguments

. . .	Arguments passed to ‘eqty_sdk.compute()‘.

Value

The Python compute decorator or decorated callable returned by the SDK.

eqty — *Direct access to the Python SDK module*

Description

‘eqty‘ is retained for compatibility. Prefer the exported R names such as [Dataset], [Context], and [integrity_init()] for new code.

Value

A list of class `eqty_sdk_proxy` representing the Python `eqty_sdk` module. Accessing a member with `$` loads the module and returns the requested Python object through **reticulate**. Python objects inherit from `python.builtin.object`; values that **reticulate** converts automatically are returned as the corresponding R values.

Examples

```
# The compatibility proxy can be inspected without initializing Python.
class(eqty)
stopifnot(inherits(eqty, "eqty_sdk_proxy"))
```

eqty-sdk-types — *Python-backed Eqty SDK Types*

Description

These names are lazy proxies for the matching ‘eqty_sdk‘ classes and enums. They are available after ‘library(eqty.sdk.r)‘ and resolve the installed Python SDK only when a member is accessed. They retain its constructors, factories, properties, and methods.

Details

The built-in asset categories are ‘Agent‘, ‘Benchmark‘, ‘BenchmarkResult‘, ‘Binary‘, ‘Certificate‘, ‘Code‘, ‘Configuration‘, ‘Credential‘, ‘Custom‘, ‘Database‘, ‘Dataset‘, ‘Document‘, ‘Guardrail‘, ‘Media‘, ‘Model‘, ‘Prompt‘, ‘Reasoning‘, ‘Skill‘, ‘SystemPrompt‘, ‘Token‘, and ‘Tool‘. They share a common construction pattern. Use ‘with_context(ctx)‘ when the asset belongs to a specific context.

[CID] identifies content and is used to connect computation inputs and outputs. [UUID] identifies objects whose identity is not derived from their content. [DID] identifies signers and other entities referenced by provenance statements.

[Service] connects the SDK to a remote Integrity Service. It is unnecessary when a workflow only exports a local manifest.

Value

Each exported name is a list of class `eqty_sdk_proxy` that lazily resolves the corresponding Python SDK class, enum, or constant. Accessing a member with `$` returns that member through **reticulate**. Constructors and factories return Python-backed objects inheriting from `python.builtin.object`, with SDK-specific classes such as `Dataset`, `Context`, or `CID`. These objects represent assets, provenance contexts, content identifiers, and the other SDK types described above. Enum members and constants represent the SDK's supported categories and algorithms.

Examples

```
# Inspect the lazy proxy without initializing Python.
class(Dataset)
stopifnot(inherits(Dataset, "eqty_sdk_proxy"))
```

`get_cid_for_bytes` *Compute a CID for Bytes*

Description

A CID is derived from content and changes when that content changes.

Usage

```
get_cid_for_bytes(...)
```

Arguments

`...`	Arguments passed to 'eqty_sdk.get_cid_for_bytes()'.

Value

A [CID] for the supplied bytes.

`get_cid_for_json` *Compute a CID for JSON*

Description

Compute a CID for JSON

Usage

```
get_cid_for_json(...)
```

Arguments

...	Arguments passed to ‘eqty_sdk.get_cid_for_json()‘.

Value

A [CID] for the supplied JSON value.

---

get_cid_for_path *Compute a CID for a File or Directory*

Description

Compute a CID for a File or Directory

Usage

```
get_cid_for_path(...)
```

Arguments

...	Arguments passed to ‘eqty_sdk.get_cid_for_path()‘.

Value

A [CID] derived from the content at the supplied path.

---

integrity_compute *Wrap an R Function as a Computation*

Description

Creates a callable computation from an R function. When the returned function is called, the SDK records its inputs and outputs in the selected [Context]. The function executes directly in R. Its source is registered as a [Code] asset, and each call creates a [Computation] that connects that code and the call's inputs to its outputs. The computation can then serve as an entry point when the context is exported as a manifest.

Usage

```
integrity_compute(
  context = NULL,
  metadata = list(),
  func = NULL,
  `_store` = FALSE
)
```

Arguments

`context`	An optional [Context] created with ‘Context$new()‘ or a context factory. When omitted, the default context configured by [integrity_init()] is used.
`metadata`	A named list of metadata for the computation, such as ‘list(name = "My algorithm", version = "1.0")‘. The ‘name‘ value is also used as the display name of the recorded R code asset. If omitted, it defaults to ‘"R Function"‘.
`func`	The R function to wrap. When ‘NULL‘, the result is a function that accepts the R function to wrap.
`_store`	Whether to store the wrapped R function’s source content in the local blob store. Defaults to ‘FALSE‘. Set this to ‘TRUE‘ only when the source is safe to include and contains no secrets.

Value

An R function that calls ‘func‘ and records the computation’s inputs and outputs. Calling this function returns a Python-backed SDK asset inheriting from `python.builtin.object`, or an R list of such assets for multiple outputs. Each asset represents a result of ‘func‘ and provides its content identifier through ‘$cid‘. If ‘func‘ is ‘NULL‘, returns an R function that accepts the function to wrap; creating this intermediate function does not initialize Python.

Examples

```
# Create a decorator without requiring Python or the SDK.
sum_decorator <- integrity_compute(metadata = list(name = "Sum Generator"))
stopifnot(is.function(sum_decorator))

## Not run:
# Requires Python >= 3.10 and the 'eqty-sdk' Python package.
# First use may download and install this additional software.
ctx <- Context$new("Example computation")
invisible(integrity_init(default_context = ctx, custom_dir = tempdir()))
signer <- Signer$load_or_create(name = "example-computation-signer")
invisible(integrity_set_active_signer(signer))

my_algo <- integrity_compute(
  context = ctx,
  metadata = list(name = "Sum Generator"),
  func = function(x, y) x + y
)

# The result is an SDK asset representing the sum.
result <- my_algo(10, 20)
result$cid

## End(Not run)
```

integrity_dict — *Create a Python Dictionary for SDK Calls*

Description

Converts named R values to a Python ‘dict‘ for SDK methods that expect one.

Usage

```
integrity_dict(...)
```

Arguments

...	Named values and optional arguments passed to [reticulate::dict()].

Value

A Python ‘dict‘ object.

integrity_init — *Initialize the Eqty SDK*

Description

Prepares the SDK’s local state. Call this before changing configuration or creating lineage statements. Supply a default context when assets and computations should use it unless another context is selected explicitly.

Usage

```
integrity_init(...)
```

Arguments

...	Arguments passed to ‘eqty_sdk.init()‘, including ‘default_context‘ and ‘custom_dir‘.

Value

A Python-backed [Config] object for hashing, blob storage, directory filtering, and default-context settings.

integrity_path *Create a Python Path for SDK Calls*

Description

Converts an R path value to Python's ‘pathlib.Path‘, as required by methods such as ‘Context$export()‘ and ‘Context$import_manifest()‘.

Usage

```
integrity_path(path)
```

Arguments

path	A character string or path-like value.

Value

A Python ‘pathlib.Path‘ object.

integrity_resolve_path *Resolve a Path from an SDK Value*

Description

Extracts a character path from an SDK asset, a Python path object, or a character value. If no direct path is available, the helper also checks the object's value.

Usage

```
integrity_resolve_path(input_object)
```

Arguments

input_object	An SDK asset, Python path object, or character value.

Value

A character string containing the resolved path. Returns ‘NULL‘ if no path could be determined.

Examples

```
integrity_resolve_path("data/input.csv")
integrity_resolve_path(list(path = "output/result.csv"))
stopifnot(identical(
  integrity_resolve_path("data/input.csv"), "data/input.csv"
))
```

integrity_set_active_signer *Set the Active Signer*

Description

Sets the signer used when higher-level SDK operations create signed statements. Configure it during workflow setup after [integrity_init()].

Usage

```
integrity_set_active_signer(signer)
```

Arguments

signer	A signer created by [Signer].

Value

‘NULL‘.

purge_blob_store *Clear the Local Blob Store*

Description

Removes blobs cached in the SDK’s local state. This is intended primarily for testing, development, or an intentional local reset.

Usage

```
purge_blob_store(...)
```

Arguments

...	Arguments passed to ‘eqty_sdk.purge_blob_store()‘.

Value

‘NULL‘.

purge_statement_store *Clear the Local Statement Store*

Description

Removes locally recorded lineage statements. Normal application workflows generally preserve this state.

Usage

```
purge_statement_store(...)
```

Arguments

...	Arguments passed to ‘eqty_sdk.purge_statement_store()‘.

Value

‘NULL‘.

verify_statement *Verify a Lineage Statement*

Description

Recomputes the statement identifier from its canonicalized JSON-LD content and checks it against the statement’s recorded identifier. The check runs locally. Pass the manifest’s context map when custom JSON-LD contexts are involved.

Usage

```
verify_statement(...)
```

Arguments

...	Arguments passed to ‘eqty_sdk.verify_statement()‘.

Value

‘TRUE‘ when the statement identifier matches its content; otherwise ‘FALSE‘.

`verify_vc` *Verify a Verifiable Credential*

Description

Checks a W3C Verifiable Credential proof locally. A statement identifier can also be supplied to confirm that the credential refers to the expected statement. Revocation and suspension status are outside this offline check.

Usage

```
verify_vc(...)
```

Arguments

...	Arguments passed to ‘eqty_sdk.verify_vc()‘.

Value

‘TRUE‘ when the credential proof and optional subject binding verify; otherwise ‘FALSE‘.

Index

Agent *(eqty-sdk-types)*, 3
Asset *(eqty-sdk-types)*, 3
AssetType *(eqty-sdk-types)*, 3
Association *(eqty-sdk-types)*, 3
ASSOCIATION_TYPES *(eqty-sdk-types)*, 3

Benchmark *(eqty-sdk-types)*, 3
BenchmarkResult *(eqty-sdk-types)*, 3
Binary *(eqty-sdk-types)*, 3

Certificate *(eqty-sdk-types)*, 3
CID *(eqty-sdk-types)*, 3
Code *(eqty-sdk-types)*, 3
Computation *(eqty-sdk-types)*, 3
Compute *(eqty-sdk-types)*, 3
compute, 2
Config *(eqty-sdk-types)*, 3
Configuration *(eqty-sdk-types)*, 3
Context *(eqty-sdk-types)*, 3
Credential *(eqty-sdk-types)*, 3
Custom *(eqty-sdk-types)*, 3

Database *(eqty-sdk-types)*, 3
Dataset *(eqty-sdk-types)*, 3
Declaration *(eqty-sdk-types)*, 3
DID *(eqty-sdk-types)*, 3
Document *(eqty-sdk-types)*, 3

Entity *(eqty-sdk-types)*, 3
eqty, 3
eqty-sdk-types, 3
Error *(eqty-sdk-types)*, 3

get_cid_for_bytes, 4
get_cid_for_json, 4
get_cid_for_path, 5
Guardrail *(eqty-sdk-types)*, 3

integrity_compute, 5
integrity_dict, 7
integrity_init, 7
integrity_path, 8
integrity_resolve_path, 8
integrity_set_active_signer, 9

Media *(eqty-sdk-types)*, 3
Model *(eqty-sdk-types)*, 3

Prompt *(eqty-sdk-types)*, 3
purge_blob_store, 9
purge_statement_store, 10

Reasoning *(eqty-sdk-types)*, 3

Service *(eqty-sdk-types)*, 3
Signer *(eqty-sdk-types)*, 3
SIGNER_ALGORITHMS *(eqty-sdk-types)*, 3
Skill *(eqty-sdk-types)*, 3
SystemPrompt *(eqty-sdk-types)*, 3

Token *(eqty-sdk-types)*, 3
Tool *(eqty-sdk-types)*, 3

UsageError *(eqty-sdk-types)*, 3
UUID *(eqty-sdk-types)*, 3

verify_statement, 10
verify_vc, 11