

# Package: neptune (via r-universe)

October 26, 2024

**Type** Package

**Title** MLOps Metadata Store - Experiment Tracking and Model Registry for Production Teams

**Version** 0.3.0

**Maintainer** Rafal Jankowski <rafal.jankowski@neptune.ai>

**Description** An interface to Neptune. A metadata store for MLOps, built for teams that run a lot of experiments. It gives you a single place to log, store, display, organize, compare, and query all your model-building metadata. Neptune is used for: • Experiment tracking: Log, display, organize, and compare ML experiments in a single place. • Model registry: Version, store, manage, and query trained models, and model building metadata. • Monitoring ML runs live: Record and monitor model training, evaluation, or production runs live For more information see <<https://neptune.ai/>>.

**License** Apache License 2.0 | file LICENSE

**URL** <https://github.com/neptune-ai/neptune-r>

**BugReports** <https://github.com/neptune-ai/neptune-r/issues>

**Encoding** UTF-8

**SystemRequirements** Python (>= 3.0.0), Neptune (<https://neptune.ai/>)

**Imports** reticulate (>= 1.22), this.path (>= 0.4.4), rstudioapi (>= 0.7), ggplot2 (>= 3.3.5), plotly (>= 4.9.4.1), htmlwidgets (>= 1.5.3), methods

**Suggests** testthat (>= 3.0.0)

**Repository** <https://neptune-ai.r-universe.dev>

**RemoteUrl** <https://github.com/neptune-ai/neptune-r>

**RemoteRef** HEAD

**RemoteSha** 7c51ab3da976dcc9407999a6a3d5b8f8c6eb714f

## Contents

ANONYMOUS_API_TOKEN . . . . .	2
neptune_add . . . . .	3
neptune_append . . . . .	4
neptune_assign . . . . .	5
neptune_clear . . . . .	6
neptune_delete_files . . . . .	7
neptune_download . . . . .	8
neptune_download_last . . . . .	9
neptune_exists . . . . .	10
neptune_extend . . . . .	11
neptune_fetch . . . . .	13
neptune_fetch_files_list . . . . .	14
neptune_fetch_hash . . . . .	15
neptune_fetch_last . . . . .	16
neptune_fetch_runs_table . . . . .	17
neptune_fetch_values . . . . .	18
neptune_file_as_html . . . . .	19
neptune_file_as_image . . . . .	20
neptune_get_structure . . . . .	21
neptune_get_url . . . . .	22
neptune_init_run . . . . .	23
neptune_install . . . . .	26
neptune_pop . . . . .	27
neptune_print_structure . . . . .	28
neptune_remove . . . . .	29
neptune_set_api_token . . . . .	30
neptune_stop . . . . .	31
neptune_sync . . . . .	32
neptune_track_files . . . . .	33
neptune_upload . . . . .	34
neptune_upload_files . . . . .	35
neptune_wait . . . . .	36
[.neptune.new.metadata_containers.run.Run . . . . .	37
[<-.neptune.new.metadata_containers.run.Run . . . . .	38
<b>Index</b>	<b>40</b>

---

ANONYMOUS\_API\_TOKEN    *API token for anonymous logging.*

---

## Description

You can use this value for the `api_token` parameter of the `init` methods.

**Author(s)**

Authors:

- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:  
# We are using api token for an anonymous user neptuner. For your projects use your private token.  
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,  
                    project = "common-r/quickstarts")  
  
## End(Not run)
```

---

neptune_add	<i>Adds the provided tag or tags to the run's tags.</i>
-------------	---

---

**Description**

Adds the provided tag or tags to the run's tags.

**Usage**

```
neptune_add(x, values, wait=FALSE)
```

**Arguments**

x	Field for which we want to add tags eg. run['sys/tags']
values	string or list of strings - Tag or tags to be added.
wait	logical - If TRUE the client will wait to send all tracked metadata to the server before making the assignment. This makes the call synchronous.

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:
# We are using api token for an anonymous user neptuner. For your projects use your private token.
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                   project = "common-r/quickstarts")
neptune_add(run['sys/tags'], 'lgbm')

## End(Not run)
```

---

neptune_append	<i>Appends the provided number or a collection of numbers.</i>
----------------	--

---

**Description**

Logs the provided number or a collection of numbers. Available for following field types (<https://docs.neptune.ai/api-reference/field-types>): \* 'FloatSeries' \* 'StringSeries' \* 'FileSeries'

**Usage**

```
neptune_append(x, value, step=NULL, timestamp=NULL, wait=FALSE, ...)
```

**Arguments**

x	Field, for which we want to append value
value	Value or collection of values to be added to the field.
step	(numeric or integer) - Index of the log entry being appended. Must be strictly increasing.
timestamp	(numeric or integer) - Time index of the log entry being appended in form of Unix time. If NULL current time will be used as a timestamp.
wait	logical - If TRUE the client will wait to send all tracked metadata to the server before making the assignment. This makes the call synchronous.
...	Other arguments passed on to the ggsave function in case of logging a ggplot2 object.

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:  
# We are using api token for an anonymous user neptuner. For your projects use your private token.  
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,  
                   project = "common-r/quickstarts")  
neptune_append(run['loss'], 0.1)  
  
## End(Not run)
```

---

neptune_assign	<i>Assigns the provided value to the field.</i>
----------------	---

---

**Description**

Assigns the provided value to the field.

**Usage**

```
neptune_assign(x, value, wait=FALSE)
```

**Arguments**

x	Field or run, for which we want to assign the value
value	Value to be stored in a field.
wait	logical - If TRUE the client will wait to send all tracked metadata to the server. This makes the call synchronous.

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:  
# We are using api token for an anonymous user neptuner. For your projects use your private token.  
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,  
                   project = "common-r/quickstarts")  
neptune_assign(run['parameters'], list(epochs=100, lr=0.01))  
  
## End(Not run)
```

---

neptune_clear	<i>Removes all tags from the StringSet.</i>
---------------	---

---

**Description**

Removes all tags from the StringSet.

**Usage**

```
neptune_clear(x, wait=FALSE)
```

**Arguments**

x	Field
wait	logical - If TRUE the client will wait to send all tracked metadata to the server before making the assignment. This makes the call synchronous.

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
  project = "common-r/quickstarts")
neptune_clear(run['sys/tags'])

## End(Not run)
```

---

neptune\_delete\_files *Delete the file or files specified by paths from the FileSet stored on the Neptune servers.*

---

**Description**

Delete the file or files specified by paths from the FileSet stored on the Neptune servers.

**Usage**

```
neptune_delete_files(x, paths, wait=FALSE)
```

**Arguments**

x	Field
paths	character or vector of characters - Path or paths to files or folders to be deleted. Note that these are paths relative to the FileSet itself e.g. if the FileSet contains file example.txt, varia/notes.txt, varia/data.csv to delete whole subfolder you would pass varia as the argument.
wait	logical - If TRUE the client will wait to send all tracked metadata to the server before making the assignment. This makes the call synchronous.

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
  project = "common-r/quickstarts")
neptune_delete_files(run['artifacts/images'], "path/to/file")

## End(Not run)
```

---

neptune_download	<i>Downloads all the files that are referenced in the field.</i>
------------------	--

---

**Description**

Downloads all the files that are referenced in the field.

**Usage**

```
neptune_download(x, destination)
```

**Arguments**

x	Field for which we want to download the files
destination	(character - The directory where the files will be downloaded. If NULL is passed, the files will be downloaded to the current working directory. If destination is a directory, the file will be downloaded to the specified directory with a filename composed from field name and extension (if present). If destination is a path to a file, the file will be downloaded under the specified name.

**Value**

None



**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:
# We are using api token for an anonymous user neptuner. For your projects use your private token.
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                   project = "common-r/quickstarts",
                   run='AR-2', # Neptune Run ID of a run with artifact
                   mode='read-only')
neptune_download(run['artifacts/images'], destination='datasets/train/images')

## End(Not run)
```

---

neptune\_download\_last *Downloads the last File stored in the series from Neptune servers and save it locally.*

---

**Description**

Downloads the last File stored in the series from Neptune servers and save it locally.

**Usage**

```
neptune_download_last(x, destination=NULL)
```

**Arguments**

x	Field
destination	character - The directory where the file will be downloaded. If NULL is passed, the file will be downloaded to the current working directory. If destination is a directory, the file will be downloaded to the specified directory with a filename composed from field name and extension (if present). If destination is a path to a file, the file will be downloaded under the specified name.

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                   project = "common-r/quickstarts")
df <- neptune_download_last(run['train/predictions'])

## End(Not run)
```

---

neptune\_exists

*Checks if there is any field or namespace under the specified path.*


---

**Description**

Checks if there is any field or namespace under the specified path. Note that this method checks the local representation of the run. The field may have been created by another process (use `neptune_sync(run)` to synchronize local representation) or the metadata may have not reached the Neptune servers so it may be impossible to fetch (use `neptune_wait(run)` to wait for all tracking calls to finish).

**Usage**

```
neptune_exists(run, path)
```

**Arguments**

run	Neptune run object.
path	character - The path to check for the existence of a field or a namespace

**Value**

logical - TRUE if field exists at path

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:  
# We are using api token for an anonymous user neptuner.  
# For your projects use your private token.  
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,  
                    project = "common-r/quickstarts")  
run['lr'] <- 0.001  
neptune_exists(run, 'lr')  
neptune_exists(run, 'notlr')  
  
## End(Not run)
```

---

neptune_extend	<i>Extends the field with collection of values.</i>
----------------	---

---

**Description**

Extends the field with collection of values. Available for following field types (<https://docs.neptune.ai/api-reference/field-types>): \* 'FloatSeries' \* 'StringSeries' \* 'FileSeries'

**Usage**

```
neptune_extend(x, values, steps=NULL, timestamps=NULL, wait=FALSE, ...)
```

**Arguments**

x	Field, for which we want to append value
values	Collection of values to be added to the field.
steps	Collection of steps (numeric or integer) - Indexes of the log entries being extended. Must be strictly increasing.
timestamps	(numeric or integer) - Time indexes of the log entries being extended in form of Unix time. If NULL current time will be used as a timestamp.
wait	logical - If TRUE the client will wait to send all tracked metadata to the server before making the assignment. This makes the call synchronous.
...	Other arguments passed on to the ggsave function in case of logging a ggplot2 object.

**Value**

None

**Author(s)**

Authors:

- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:  
# We are using api token for an anonymous user neptuner. For your projects use your private token.  
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,  
                   project = "common-r/quickstarts")  
neptune_extend(run['loss'], c(0.1, 0.2, 0.3))  
  
## End(Not run)
```

---

neptune_fetch	<i>Fetch values of all non-File Atom fields as a named list.</i>
---------------	--

---

### Description

Fetch values of all non-File Atom fields as a named list. The result will preserve the hierarchical structure of the run's metadata, but will contain only non-File Atom fields. You can use this method to quickly retrieve previous run's parameters.

### Usage

```
neptune_fetch(x)
```

### Arguments

x                   Field or run, for which we want to fetch the value

### Value

named list containing all non-File Atom fields values.

### Author(s)

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

### See Also

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

### Examples

```
## Not run:
# We are using api token for an anonymous user neptuner. For your projects use your private token.
resumed_run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                           project = "common-r/quickstarts",
                           run="HEL-3")
params <- neptune_fetch(resumed_run['model/paramaters'])
run_data <- neptune_fetch(resumed_run)
print(run_data)
# this will print out all Atom attributes stored in run as a dict

## End(Not run)
```

---

`neptune_fetch_files_list`*Fetches a list of artifact files from the Neptune servers.*

---

## Description

Fetches a list of artifact files from the Neptune servers.

## Usage

```
neptune_fetch_files_list(x)
```

## Arguments

x	Field
---	-------

## Value

List of `ArtifactFileData` objects for all the files referenced in the artifacts. `ArtifactFileData` has the following fields that you can use: `file_hash`: str, Hash of the file `file_path`: str, URL of the file in the Neptune UI `size`: int, Size of the file in KB `metadata`: dict, dictionary with the following keys: `file_path`: a location (path) of the file either on local storage or S3-compatible storage `last_modified`: when was the last the artifact content was changed

## Author(s)

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

## See Also

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

## Examples

```
## Not run:
run <- neptune_init(project='<YOUR_WORKSPACE/YOUR_PROJECT>',
  api_token='<YOUR_API_TOKEN>',
  run='AR-2', # Neptune Run ID of a run with artifact
  mode='read-only')
artifact_list <- neptune_fetch_files_list(run['artifacts/images'])
artifact_list[[1]]$file_hash
```

```
artifact_list[[1]]$file_path
artifact_list[[1]]$metadata['last_modified']

## End(Not run)
```

---

neptune\_fetch\_hash      *Fetches the Hash of the artifact from Neptune servers.*

---

## Description

Fetches the Hash of the artifact from Neptune servers.

## Usage

```
neptune_fetch_hash(x)
```

## Arguments

x                      Field

## Value

Hash of a Neptune artifact as a character.

## Author(s)

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

## See Also

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

## Examples

```
## Not run:
# We are using api token for an anonymous user neptuner.
# For your projects use your private token.
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                    project = "common-r/quickstarts")
neptune_fetch_hash(run['artifacts/images'])

## End(Not run)
```

---

neptune\_fetch\_last      *Fetches last value stored in the series from Neptune servers.*

---

### Description

Fetches last value stored in the series from Neptune servers.

### Usage

```
neptune_fetch_last(x)
```

### Arguments

x                      Field

### Value

Last float value logged.

### Author(s)

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

### See Also

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

### Examples

```
## Not run:  
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,  
                   project = "common-r/quickstarts")  
last_auc_value <- neptune_fetch_last(run['metrics/auc'])  
  
## End(Not run)
```



---

`neptune_fetch_runs_table`*Retrieve runs matching the specified criteria.*

---

### Description

Retrieve runs matching the specified criteria. All parameters are optional, each of them specifies a single criterion. Only runs matching all of the criteria will be returned. Due to technical limitation only first 10,000 runs matching the criteria are fetched.

### Usage

```
neptune_fetch_runs_table(project, id = NULL, state = NULL, owner = NULL, tag = NULL)
```

### Arguments

<code>project</code>	string - project name
<code>id</code>	(string or vector of strings, optional, default is NULL) - A run's id like "SAN-1" or list of ids like <code>c("SAN-1", "SAN-2")</code> . Matching any element of the list is sufficient to pass the criterion.
<code>state</code>	(string or vector of strings, optional, default is NULL) - A run's state like "active" or list of states like <code>c("inactive", "active")</code> . Possible values: "inactive", "active". Matching any element of the list is sufficient to pass the criterion.
<code>owner</code>	(string or vector of strings, optional, default is NULL) - Username of the run's owner (the user who created the tracked run is an owner) like "josh" or a list of owners like <code>c("frederic", "josh")</code> . Matching any element of the list is sufficient to pass the criterion.
<code>tag</code>	(string or vector of strings, optional, default is NULL) - An experiment tag like "lightGBM" or list of tags like <code>c("pytorch", "cycleLR")</code> . Only experiments that have all specified tags will match this criterion.

### Value

RunsTable data in the form of DataFrame.

### Author(s)

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

## See Also

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

## Examples

```
## Not run:
# We are using api token for an anonymous user neptuner. For your projects use your private token.
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                   project = "common-r/quickstarts")
df <- neptune_fetch_runs("common-r/quickstarts")

## End(Not run)
```

---

neptune\_fetch\_values *Fetches all values stored in the series from Neptune servers.*

---

## Description

Fetches all values stored in the series from Neptune servers.

## Usage

```
neptune_fetch_values(x, include_timestamp=TRUE)
```

## Arguments

x	Field
include_timestamp	logical - Whether to include the fetched data should include the timestamp field.

## Value

data.frame containing all the values and their indexes stored in the series field.

## Author(s)

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

## See Also

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

## Examples

```
## Not run:
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                   project = "common-r/quickstarts")
df <- neptune_fetch_values(run['metrics/auc'])

## End(Not run)
```

---

neptune\_file\_as\_html *Converts an object to an HTML File value object.*

---

## Description

Converts an object to an HTML File value object. This way you can upload ggplot2, Plotly interactive charts or upload directly DataFrame objects to explore them in Neptune UI.

## Usage

```
neptune_file_as_html(x)
```

## Arguments

x An object to be converted. Supported are ggplot2, Plotly and DataFrame objects. R matrices must be converted to numpy arrays via `reticulate::np_array` R dataframes are automatically converted to Pandas

## Value

None

## Author(s)

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<[backend@neptune.ai](mailto:backend@neptune.ai)>

## See Also

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

## Examples

```
## Not run:
# We are using api token for an anonymous user neptuner. For your projects use your private token.
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                   project = "common-r/quickstarts")
df <- data.frame(prediction = runif(10), index = 1:10)
neptune_upload(run['evaluation/predictions'], neptune_file_as_html(df))

## End(Not run)
```

---

neptune\_file\_as\_image *Static method for converting image objects or image-like objects to an image File value object.*

---

## Description

Static method for converting image objects or image-like objects to an image File value object. This way you can upload ggplot2 objects, matrices, as static images.

## Usage

```
neptune_file_as_image(x, ...)
```

## Arguments

x	In docs there is Image-like object to be converted. Supported are ggplot2 objects, matrices, as static images. R matrices must be converted to numpy arrays via <code>reticulate::np_array</code>
...	Other arguments passed on to the ggsave function

## Value

None

## Author(s)

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<[backend@neptune.ai](mailto:backend@neptune.ai)>

## See Also

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

## Examples

```
## Not run:  
# We are using api token for an anonymous user neptuner. For your projects use your private token.  
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,  
                   project = "common-r/quickstarts")  
image <- matrix(runif(16*16), nrow = 16)  
neptune_log(run['test/sample_images'], neptune_file_as_image(reticulate::np_array(image)))  
  
## End(Not run)
```

---

`neptune_get_structure` Returns a run's metadata structure in form of a named list.

---

## Description

Returns a run's metadata structure in form of a named list. This method can be used to traverse the run's metadata structure programmatically when using Neptune in automated workflows.

## Usage

```
neptune_get_structure(run)
```

## Arguments

`run` Neptune run object.

## Value

Named list containing all metadata stored for run

## Author(s)

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<[backend@neptune.ai](mailto:backend@neptune.ai)>

## See Also

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

## Examples

```
## Not run:  
# We are using api token for an anonymous user neptuner. For your projects use your private token.  
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,  
                   project = "common-r/quickstarts")  
str <- neptune_get_structure(run)  
  
## End(Not run)
```

---

neptune_get_url	<i>Returns a direct link to run in Neptune. It's the same link that is printed at the moment of initialization of the run.</i>
-----------------	--

---

## Description

Returns a direct link to run in Neptune. It's the same link that is printed at the moment of initialization of the run.

## Usage

```
neptune_get_url(run)
```

## Arguments

run            Neptune run object.

## Value

character - URL of the run in Neptune

## Author(s)

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<[backend@neptune.ai](mailto:backend@neptune.ai)>

## See Also

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

## Examples

```
## Not run:
# We are using api token for an anonymous user neptuner.
# For your projects use your private token.
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                   project = "common-r/quickstarts")
neptune_get_url(run)

## End(Not run)
```

---

neptune_init_run	<i>Starts a new tracked run, and append it to the top of the Runs table view.</i>
------------------	---

---

## Description

Starts a new tracked run, and append it to the top of the Runs table view. All parameters are optional, hence minimal invocation: `run <- neptune_init()`.

## Usage

```
neptune_init(project = NULL,
             api_token = NULL,
             run = NULL,
             python = NULL,
             python_path,
             source_files = NULL,
             mode = "async",
             custom_run_id = NULL,
             name = NULL,
             description = NULL,
             tags = NULL,
             capture_stdout = TRUE,
             capture_stderr = TRUE,
             capture_hardware_metrics = TRUE,
             capture_traceback = TRUE,
             monitoring_namespace = NULL,
             fail_on_exception = TRUE,
             flush_period = 5,
             proxies = NULL)
```

**Arguments**

project	character – Name of a project in a form of namespace/project_name. If NULL, the value of NEPTUNE_PROJECT environment variable will be taken.
api_token	character – User’s API token. If NULL, the value of NEPTUNE_API_TOKEN environment variable will be taken. Note: It is strongly recommended to use NEPTUNE_API_TOKEN environment variable rather than placing your API token in plain text in your source code.
run	character – An existing run’s identifier like ‘SAN-1’ in case of resuming a tracked run. A run with such identifier must exist. If NULL is passed, starts a new tracked run. See Resume run guide.
python	character - one of python, conda, miniconda, venv. Specifies which type of python’s backend should be used.
python_path	character - path to python executable.
source_files	character vector – List of source files to be uploaded. Must be list of str or single str. Uploaded sources are displayed in the run’s Source code tab. If NULL is passed, for non-interactive sessions the R script from which the method is called will be uploaded. Unix style pathname pattern expansion is supported. For example, you can pass ‘*.py’ to upload all python source files from the current directory. For Python 3.5 or later, paths of uploaded files on server are resolved as relative to the calculated common root of all uploaded source files. For older Python versions, paths on server are resolved always as relative to the current directory. For recursion lookup use ‘**/*.py’ (for Python 3.5 and later). For more information see glob library.
mode	character – Connection mode in which the tracking will work. Possible values "async", "sync", "offline", and "debug", "read-only". See Connection modes guide.
custom_run_id	character – A unique identifier that can be used to log metadata to a single run from multiple files or processes. The maximum length of the identifier is 32 characters. Make sure you are using the same identifier everywhere. See Pipelines guide. If NULL and the NEPTUNE_CUSTOM_RUN_ID environment variable is set Neptune will use its value as custom_run_id.
name	character – Editable name of the run. Name is displayed in the run’s Details and in Runs table as a column.
description	character – Editable description of the run. Description is displayed in the run’s Details and can be displayed in the runs view as a column.
tags	character vector – Tags of the run. They are editable after run is created. Tags are displayed in the run’s Details and can be viewed in Runs table view as a column.
capture_stdout	logical – Whether to send python’s stdout. Tracked metadata will be stored inside monitoring_namespace.
capture_stderr	logical – Whether to send python’s stderr. Tracked metadata will be stored inside monitoring_namespace.



capture_hardware_metrics	logical – Whether to send hardware monitoring logs (CPU, GPU, Memory utilization). Tracked metadata will be stored inside monitoring_namespace.
capture_traceback	logical – Whether to send run's traceback in case of an exception on Python's side.
monitoring_namespace	character - Namespace inside which all monitoring logs be stored.
fail_on_exception	logical – Whether to set run's as True, if an uncaught exception on python's side happens.
flush_period	numeric - how often asynchronous thread should synchronize data with Neptune servers. Value in seconds.
proxies	named list (python's dict) - Argument passed to HTTP calls made via the python's Requests library. For more information see their proxies section.

**Value**

Run object that is used to manage the tracked run and log metadata to it.

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:
# We are using api token for an anonymous user neptuner. For your projects use your private token.
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                    project = "common-r/quickstarts")

## End(Not run)
```

---

neptune_install	<i>Install neptune-client along with required python enviroment.</i>
-----------------	--

---

### Description

neptune\_install() installs just the neptune-client python package and it's direct dependencies.

### Usage

```
neptune_install(method = c("auto", "virtualenv", "conda"),
  conda = "auto",
  version = "0.16",
  envname = NULL,
  extra_packages = c("psutil"),
  restart_session = TRUE,
  conda_python_version = "3.7",
  ...,
  python_version = conda_python_version)
```

### Arguments

method	Installation method. By default, "auto" automatically finds a method that will work in the local environment. Change the default to force a specific installation method. Note that the "virtualenv" method is not available on Windows.
conda	The path to a conda executable. Use "auto" to allow reticulate to automatically find an appropriate conda binary. See Finding Conda for more details.
version	Neptune-client version to install. Valid values include: + "default" installs 'r default_version' + "release" installs the latest release version of neptune-client + A version specification like "0.9" or "0.9.16". Note that if the patch version is not supplied, the latest patch release is installed + The full URL or path to a installer binary or python *.whl file.
envname	The name, or full path, of the environment in which Python packages are to be installed. When NULL (the default), the active environment as set by the RETICULATE_PYTHON_ENV variable will be used; if that is unset, then the r-reticulate environment will be used.
extra_packages	Additional Python packages to install along with Neptune
restart_session	Restart R session after installing (note this will only occur within RStudio).
conda_python_version	python_version,conda_python_version the python version installed in the created conda environment. Ignored when attempting to install with a Python virtual environment.
...	other arguments passed to [reticulate::conda_install()] or [reticulate::virtualenv_install()], depending on the 'method' used. Accepts 'pip_ignore_installed=TRUE'.

`python_version` The requested Python version. Ignored when attempting to install with a Python virtual environment.

### Author(s)

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

### See Also

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

### Examples

```
## Not run:
neptune_install()

## End(Not run)
```

---

<code>neptune_pop</code>	<i>Removes the field or whole namespace stored under the path completely and all data associated with them.</i>
--------------------------	---

---

### Description

Removes the field or whole namespace stored under the path completely and all data associated with them.

### Usage

```
neptune_pop(run, path, wait=TRUE)
```

### Arguments

<code>run</code>	Neptune run object.
<code>path</code>	character - Path of the field or namespace to be removed.
<code>wait</code>	logical - If TRUE the client will first wait to send all tracked metadata to the server. This makes the call synchronous, see Connection modes guide.

### Value

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:
# We are using api token for an anonymous user neptuner.
# For your projects use your private token.
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                    project = "common-r/quickstarts")
run['lr'] <- 0.001
neptune_pop(run, 'lr')

## End(Not run)
```

---

neptune\_print\_structure

*Pretty prints the structure of the run's metadata. Paths are ordered lexicographically and the whole structure is neatly colored.*

---

**Description**

Pretty prints the structure of the run's metadata. Paths are ordered lexicographically and the whole structure is neatly colored.

**Usage**

```
neptune_print_structure(run)
```

**Arguments**

run                    Neptune run object.

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:
# We are using api token for an anonymous user neptuner. For your projects use your private token.
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                    project = "common-r/quickstarts")
neptune_print_structure(run)

## End(Not run)
```

---

neptune_remove	<i>Removes the provided tag or tags from the set.</i>
----------------	---

---

**Description**

Removes the provided tag or tags from the set.

**Usage**

```
neptune_remove(x, values, wait=FALSE)
```

**Arguments**

x	Field, for which we want to remove tags eg. run['Sys/tags']
values	string or list of strings - Tag or tags to be removed.
wait	logical - If TRUE the client will wait to send all tracked metadata to the server before making the assignment. This makes the call synchronous.

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:
# We are using api token for an anonymous user neptuner. For your projects use your private token.
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                   project = "common-r/quickstarts")
neptune_add(run['Sys/tags'], 'some_tag')
neptune_remove(run['Sys/tags'], 'some_tag')

## End(Not run)
```

---

neptune\_set\_api\_token *Sets NEPTUNE\_API\_TOKEN environment variables*

---

**Description**

Sets NEPTUNE\_API\_TOKEN environment variables. It is used as a default api\_token value by neptune\_init.

**Usage**

```
neptune_set_api_token(token)
```

**Arguments**

token                    string - Your Neptune API token

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
neptune_set_api_token("ANONYMOUS")
```

---

neptune_stop	<i>Stop neptune run.</i>
--------------	--------------------------

---

**Description**

Stop neptune run.

**Usage**

```
neptune_stop(run, seconds=NULL)
```

**Arguments**

run	Neptune run object.
seconds	(integer or numeric, optional, default is NULL) - The method will wait for the specified time for all tracking calls to finish, before stopping the connection. If NULL will wait for all tracking calls to finish.

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:  
# We are using api token for an anonymous user neptuner. For your projects use your private token.  
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,  
                   project = "common-r/quickstarts")  
neptune_stop(run)  
  
## End(Not run)
```

---

neptune\_sync

*Synchronizes the run with with Neptune servers.*

---

**Description**

Synchronizes the run with Neptune servers.

**Usage**

```
neptune_sync(run, wait=TRUE)
```

**Arguments**

run	Neptune run object.
wait	logical - If TRUE the client will first wait to send all tracked metadata to the server. This makes the call synchronous, see Connection modes guide.

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<[backend@neptune.ai](mailto:backend@neptune.ai)>



**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:
# We are using api token for an anonymous user neptuner. For your projects use your private token.
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                   project = "common-r/quickstarts")
run['lr'] <- 0.001
neptune_sync(run)

## End(Not run)
```

---

neptune\_track\_files *Saves the artifact metadata.*

---

**Description**

Saves the artifact metadata: version (hash), location (path), size, folder structure, and contents to Neptune. Works for files, folders or S3-compatible storage.

**Usage**

```
neptune_track_files(x, path, destination=NULL, wait=FALSE)
```

**Arguments**

x	Field
path	character - File path or S3-compatible path to the file or folder that you want to track.
destination	character - location inside Neptune artifact namespace where you want to log the metadata.
wait	logical - - If TRUE the client will wait to send all tracked metadata to the server before making the assignment. This makes the call synchronous.

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:
# We are using api token for an anonymous user neptuner.
# For your projects use your private token.
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                    project = "common-r/quickstarts")
neptune_track_files(run['artifacts/images'], 'datasets/train/images')

## End(Not run)
```

---

neptune_upload	<i>Uploads provided file under specified field path</i>
----------------	---

---

**Description**

Uploads provided file under specified field path

**Usage**

```
neptune_upload(x, value, wait=FALSE, ...)
```

**Arguments**

x	Field
value	string - Path to the file to be uploaded or File value object eg. object returned by neptune_file_as_html call
wait	logical - If TRUE the client will wait to send all tracked metadata to the server before making the assignment. This makes the call synchronous.
...	Other arguments passed on to the ggsave function in case of uploading a ggplot2 object.

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:  
# We are using api token for an anonymous user neptuner. For your projects use your private token.  
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,  
                   project = "common-r/quickstarts")  
neptune_upload(run['model'], "model.RData")  
  
## End(Not run)
```

---

neptune\_upload\_files *Uploads the provided file or files and stores them inside the FileSet.*

---

**Description**

Uploads the provided file or files and stores them inside the FileSet.

**Usage**

```
neptune_upload_files(x, globs, wait=FALSE)
```

**Arguments**

x	Field
globs	character or vector of characters - Path or paths to the files to be uploaded.
wait	logical - If TRUE the client will wait to send all tracked metadata to the server before making the assignment. This makes the call synchronous.

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:  
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,  
                   project = "common-r/quickstarts")  
neptune_upload_files(run['artifacts/images'], "path/to/file")  
  
## End(Not run)
```

---

neptune\_wait

*Wait for all the tracking calls to finish.*

---

**Description**

Wait for all the tracking calls to finish. In asynchronous connection mode data (default one) is synchronized with the Neptune servers in the background and each call to upload, log etc. is not immediately executed. Wait delays the code execution until all the calls are done.

**Usage**

```
neptune_wait(run, disk_only=FALSE)
```

**Arguments**

run	Neptune run object.
disk_only	logical - If TRUE the process will only wait for data to be saved locally from memory, but will not wait for them to reach Neptune servers.

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:
# We are using api token for an anonymous user neptuner.
# For your projects use your private token.
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                   project = "common-r/quickstarts")
neptune_wait(run)

## End(Not run)
```

---

[.neptune.new.metadata\_containers.run.Run  
*Field lookup*

---

**Description**

Field lookup. You can access any run's field through a named-list-like field lookup `run[field_path]`. As well as fetch already tracked metadata - fetching parameters when resuming a run or downloading metadata for a custom analysis using functions like `neptune_fetch`

**Arguments**

x	Neptune Run object
i	string, name of the field to set

**Value**

Field

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

**See Also**

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

**Examples**

```
## Not run:
# We are using api token for an anonymous user neptuner. For your projects use your private token.
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,
                    project = "common-r/quickstarts")
run['epochs'] <- 100
run['epochs']

## End(Not run)
```

---

[<-neptune.new.metadata\_containers.run.Run

*Assigns the provided value to the field.*

---

**Description**

Assigns the provided value to the field.

**Arguments**

x	Neptune Run object
i	string, name of the field to set
value	value which is assigned to field

**Value**

None

**Author(s)**

Authors:

- Mateusz Dominiak
- Maintainer: Neptune DevTeam Dev<backend@neptune.ai>

## See Also

Useful links:

- <https://neptune.ai/>
- Documentation <https://docs.neptune.ai/>
- Report bugs at <https://github.com/neptune-ai/neptune-r/issues>

## Examples

```
## Not run:  
# We are using api token for an anonymous user neptuner. For your projects use your private token.  
run <- neptune_init(api_token = ANONYMOUS_API_TOKEN,  
                    project = "common-r/quickstarts")  
run['epochs'] <- 100  
  
## End(Not run)
```

# Index

\* **interface** 38

- [.neptune.new.metadata\_containers.run.Run, 37
  - ANONYMOUS\_API\_TOKEN, 2
- [<-.neptune.new.metadata\_containers.run.Run, 38
  - neptune\_add, 3
  - neptune\_append, 4
  - neptune\_assign, 5
  - neptune\_clear, 6
  - neptune\_delete\_files, 7
  - neptune\_download, 8
  - neptune\_download\_last, 9
  - neptune\_exists, 10
  - neptune\_extend, 11
  - neptune\_fetch, 13
  - neptune\_fetch\_files\_list, 14
  - neptune\_fetch\_hash, 15
  - neptune\_fetch\_last, 16
  - neptune\_fetch\_runs\_table, 17
  - neptune\_fetch\_values, 18
  - neptune\_file\_as\_html, 19
  - neptune\_file\_as\_image, 20
  - neptune\_get\_structure, 21
  - neptune\_get\_url, 22
  - neptune\_init\_run, 23
  - neptune\_install, 26
  - neptune\_log (neptune\_append), 4
  - neptune\_pop, 27
  - neptune\_print\_structure, 28
  - neptune\_remove, 29
  - neptune\_set\_api\_token, 30
  - neptune\_stop, 31
  - neptune\_sync, 32
  - neptune\_track\_files, 33
  - neptune\_upload, 34
  - neptune\_upload\_files, 35
  - neptune\_wait, 36
- [.neptune.new.metadata\_containers.run.Run, 37
- [<-.neptune.new.metadata\_containers.run.Run,