Package: vmxr (via r-universe)

July 2, 2026

Title VeloMetrix R Client

Version 0.1.1

Author Eric Novik [aut, cre], Juho Timonen [ctb], Generable [cph, fnd]

Maintainer Eric Novik <eric@generable.com>

Description A native R client for the VeloMetrix REST API (vmx-api). Wraps the treatment-to-simulation analysis workflow in ergonomic, pipe-friendly verbs that block-and-poll for asynchronous server jobs and return native R objects (tibbles and typed S3 objects). The package orchestrates server-side jobs; it holds no business logic of its own. Targets API 0.2.x / CLI 0.6.x.

License Apache License (>= 2)

URL https://github.com/generable/vmxr

BugReports https://github.com/generable/vmxr/issues

Depends R (>= 4.1.0)

Imports cli, curl, httr2, rlang, tibble, vctrs

Suggests arrow, dplyr, ggplot2, httptest2, jsonlite, knitr, rmarkdown, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.0.0

RoxygenNote 7.3.3

Config/pak/sysreqs libssl-dev

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

Date/Publication 2026-07-02 13:02:55 UTC

RemoteUrl https://github.com/generable/vmxr

RemoteRef HEAD

RemoteSha bbd051e249d4e53960bc87e8451576a59b1faed4

Contents

`vmx_analysis_log` *Audit / analysis log for a study*

Description

`GET /studies/{std_id}/analysis-log` — the unified newest-first event feed, auto-paginated into a tibble with a `kind` discriminator per row.

Usage

```
vmx_analysis_log(
  study,
  kind = NULL,
  event_type = NULL,
  outcome = NULL,
  severity = NULL,
```

```
  since = NULL,
  resource = NULL,
  client = vmx_client()
)
```

Arguments

study	A study id (std_...) or vmx_study.
kind	Optional kind filter.
event_type	Optional event-type filter.
outcome	Optional outcome filter.
severity	Optional severity filter.
since	Optional lower time bound: a POSIXct/Date (formatted to ISO-8601 UTC) or an ISO-8601 string.
resource	Optional resource id (or object) to scope to.
client	A vmx_client.

Value

A tibble.

---

vmx_client *Create a vmxr client*

Description

Resolves connection config with the precedence: explicit args -> VMX_API_BASE_URL / VMX_API_TOKEN environment variables -> ~/.Renviron -> a classed error. The token is stored on the object but redacted in print() and never logged.

Usage

```
vmx_client(base_url = NULL, token = NULL, ...)
```

Arguments

base_url	API base URL. Defaults to Sys.getenv("VMX_API_BASE_URL").
token	Authentik personal access token (PAT). Defaults to Sys.getenv("VMX_API_TOKEN"). Never hard-code a PAT in source.
...	Reserved for future options (timeouts, retries, user agent).

Value

An object of class vmx_client.

vmx_data_version *Fetch one data version*

Description

Fetch one data version

Usage

```
vmx_data_version(id, client = vmx_client())
```

Arguments

id	A data-version id (dv_...) or vmx_data_version.
client	A vmx_client.

Value

A vmx_data_version.

vmx_data_version_archive *Archive a data version*

Description

Archive a data version

Usage

```
vmx_data_version_archive(dv, reason = NULL, client = vmx_client())
```

Arguments

dv	A data-version id or vmx_data_version.
reason	Optional free-text reason.
client	A vmx_client.

Value

The updated vmx_data_version.

vmx_data_version_create

Create a data version

Description

Starts a format job over an explicit upload composition (`POST /datasets/{ds_id}/data-versions`). Returns the in-flight prep-status; poll it with `vmx_wait()`.

Usage

```
vmx_data_version_create(
  dataset,
  uploads,
  prior_config = NULL,
  client = vmx_client()
)
```

Arguments

`dataset`	A dataset id (`ds_...`) or `vmx_dataset`.
`uploads`	Character vector of upload ids (`upl_...`) to format over.
`prior_config`	Optional prior data-version (`dv_...`) whose config seeds this job (the config-update lineage pointer).
`client`	A `vmx_client`.

Value

A `vmx_prep_status` for the new job.

vmx_data_version_export

Export a data version

Description

Fetches the signed-URL export envelope (`GET /data-versions/{id}/export`). When `dest` is supplied the bundle is streamed to that path; otherwise the parsed envelope (including the signed `download_url`) is returned.

Usage

```
vmx_data_version_export(dv, dest = NULL, client = vmx_client())
```

Arguments

dv	A data-version id or vmx_data_version.
dest	Optional local file path to stream the bundle to.
client	A vmx_client.

Value

The export envelope (list), or, when dest is set, dest invisibly.

---

vmx_data_version_table

A prepared data-version table

Description

GET /data-versions/{id}/tables/{domain} — returns the formatter's prepared domain table as a tibble (columns typed per the server's column metadata, which is attached as the "columns" attribute). gen_subject_uuid is the canonical subject join key.

Usage

```
vmx_data_version_table(
  dv,
  domain = c("subjects", "pk", "dosing", "pd", "labs", "covariates"),
  client = vmx_client()
)
```

Arguments

dv	A data-version id (dv_...) or vmx_data_version.
domain	One of "subjects", "pk", "dosing", "pd", "labs", "covariates".
client	A vmx_client.

Value

A tibble.

vmx_data_version_unarchive *Unarchive a data version*

Description

Unarchive a data version

Usage

```
vmx_data_version_unarchive(dv, client = vmx_client())
```

Arguments

dv	A data-version id or vmx_data_version.
client	A vmx_client.

Value

The updated vmx_data_version.

vmx_data_versions *List data versions*

Description

List data versions

Usage

```
vmx_data_versions(
  treatment = NULL,
  study = NULL,
  include_archived = FALSE,
  eligible_for_modeling = NULL,
  client = vmx_client()
)
```

Arguments

treatment	Optional treatment filter.
study	Optional study filter.
include_archived	Include archived versions.
eligible_for_modeling	Optional modeling-eligibility filter.
client	A vmx_client.

Value

A tibble.

`vmx_dataset` *Fetch one dataset*

Description

Fetch one dataset

Usage

```
vmx_dataset(id, client = vmx_client())
```

Arguments

`id`	A dataset id (`ds_...`) or `vmx_dataset`.
`client`	A `vmx_client`.

Value

A `vmx_dataset`.

`vmx_dataset_cancel` *Cancel a dataset's in-flight format job*

Description

`POST /datasets/{ds_id}/cancel` — terminates the format job and mints a cancelled DataVersion.

Usage

```
vmx_dataset_cancel(dataset, client = vmx_client())
```

Arguments

`dataset`	A dataset id or `vmx_dataset`.
`client`	A `vmx_client`.

Value

The updated `vmx_prep_status`.

vmx_dataset_download *Download a dataset's files*

Description

Not implemented: the API's dataset-files listing does not expose per-file download URLs. Use `vmx_data_version_export()` to pull a curated bundle.

Usage

```
vmx_dataset_download(dataset, dest = ".", client = vmx_client())
```

Arguments

dataset	A dataset id or vmx_dataset.
dest	Destination directory.
client	A vmx_client.

Value

Not yet implemented.

vmx_dataset_files *List the files in a dataset*

Description

List the files in a dataset

Usage

```
vmx_dataset_files(dataset, client = vmx_client())
```

Arguments

dataset	A dataset id or vmx_dataset.
client	A vmx_client.

Value

A tibble.

vmx_dataset_tags *The tags on a dataset*

Description

Returns the dataset's allowlisted tag map as a two-column (`key`, `value`) tibble. Reads the tags off a `vmx_dataset` object when given one, else fetches the dataset.

Usage

```
vmx_dataset_tags(dataset, client = vmx_client())
```

Arguments

`dataset`	A dataset id or `vmx_dataset`.
`client`	A `vmx_client`.

Value

A tibble with `key` and `value` columns.

vmx_datasets *List datasets*

Description

List datasets

Usage

```
vmx_datasets(study = NULL, treatment = NULL, client = vmx_client())
```

Arguments

`study`	Optional study filter.
`treatment`	Optional treatment filter.
`client`	A `vmx_client`.

Value

A tibble.

vmx_dosing_input *Create a dosing input for a fit*

Description

POST /model-fits/{mf_id}/simulation-dosing-inputs.

Usage

```
vmx_dosing_input(fit, dosing_text, scenario_name, client = vmx_client())
```

Arguments

fit	A fit id (mf_...) or vmx_model_fit.
dosing_text	The dosing regimen text.
scenario_name	One or more scenario names.
client	A vmx_client.

Value

A vmx_dosing_input (carries dosing_input_id).

vmx_dosing_input_status *Dosing-input status*

Description

Dosing-input status

Usage

```
vmx_dosing_input_status(dosing_input, client = vmx_client())
```

Arguments

dosing_input	A dosing-input id or vmx_dosing_input.
client	A vmx_client.

Value

A vmx_dosing_input.

vmx_fit_global_estimates *Global (population) parameter estimates (tidy)*

Description

One row per parameter, with the point estimate and credible interval.

Usage

```
vmx_fit_global_estimates(fit, client = vmx_client())
```

Arguments

fit	A fit id or vmx_model_fit.
client	A vmx_client.

Value

A tibble.

vmx_fit_obs_vs_pred *Observed-vs-predicted diagnostic (tidy)*

Description

Reshapes the pk block's parallel observation arrays (subject ids, time, observed concentration, BLQ/ALOQ flags, LLOQ, ...) into a one-row-per- observation tibble. Non-columnar members — notably the predicted- concentration quantile bands — are kept on the "extra" attribute; the PD block is on "pd" and the fit id on "model_fit_id".

Usage

```
vmx_fit_obs_vs_pred(fit, client = vmx_client())
```

Arguments

fit	A fit id or vmx_model_fit.
client	A vmx_client.

Value

A tibble (one row per PK observation).

vmx_fit_subject_estimates *Subject-level parameter estimates (tidy, long)*

Description

One row per subject x parameter, with the posterior point estimate (`value`) and credible interval (`ci_lower`/`ci_upper`).

Usage

```
vmx_fit_subject_estimates(fit, client = vmx_client())
```

Arguments

`fit`	A fit id or `vmx_model_fit`.
`client`	A `vmx_client`.

Value

A tibble.

vmx_fit_vpc *Visual predictive check artifact*

Description

Returns the parsed VPC artifact (per dose-group and per-subject quantile bands over time grids). Tibble reshaping is deferred; see the package NEWS.

Usage

```
vmx_fit_vpc(fit, client = vmx_client())
```

Arguments

`fit`	A fit id or `vmx_model_fit`.
`client`	A `vmx_client`.

Value

A list (the parsed artifact).

vmx_health *Connectivity probe*

Description

Calls GET /health.

Usage

```
vmx_health(client = vmx_client())
```

Arguments

client	A vmx_client.

Value

Invisibly, a vmx_health object (status, version, api_contract_version, engine, engine_version); raises a classed error if the API is unreachable or unhealthy.

vmx_model_build *Start a model build run (optionally wait)*

Description

pd_marker uses the design's "GEN_uuid:increasing" / ":decreasing" shorthand; it is parsed into the API's {gen_uuid, direction} form.

Usage

```
vmx_model_build(
  data_version,
  time_basis,
  pd_marker = NULL,
  covariate = NULL,
  idempotency_key = NULL,
  retried_from = NULL,
  wait = FALSE,
  ...,
  client = vmx_client()
)
```

Arguments

data_version	A data-version id or vmx_data_version.
time_basis	Time basis.
pd_marker	Optional "GEN_uuid:increasing" / ":decreasing" string(s).
covariate	Optional covariate name(s).
idempotency_key	Optional idempotency key.
retried_from	Optional prior run to retry from.
wait	If TRUE, block until the run settles.
...	Polling controls forwarded to vmx_wait() when wait = TRUE (e.g. timeout, interval, progress).
client	A vmx_client.

Value

A vmx_model_build_run.

---

vmx_model_build_artifacts

Download build-run artifacts

Description

Download build-run artifacts

Usage

```
vmx_model_build_artifacts(run, dest = ".", client = vmx_client())
```

Arguments

run	A build-run id or object.
dest	Destination directory.
client	A vmx_client.

Value

Not yet implemented.

`vmx_model_build_cancel` *Cancel a build run*

Description

Cancel a build run

Usage

```
vmx_model_build_cancel(run, client = vmx_client())
```

Arguments

`run`	A build-run id or object.
`client`	A `vmx_client`.

Value

A `vmx_model_build_run`.

`vmx_model_build_events` *Build-run events (SSE stream)*

Description

Build-run events (SSE stream)

Usage

```
vmx_model_build_events(run, client = vmx_client())
```

Arguments

`run`	A build-run id or object.
`client`	A `vmx_client`.

Value

Not yet implemented.

vmx_model_build_export *Markdown export of a build run*

Description

Markdown export of a build run

Usage

```
vmx_model_build_export(run, client = vmx_client())
```

Arguments

run	A build-run id or object.
client	A vmx_client.

Value

The export markdown as a length-1 character vector.

vmx_model_build_logs *Build-run logs*

Description

Build-run logs

Usage

```
vmx_model_build_logs(run, client = vmx_client())
```

Arguments

run	A build-run id or object.
client	A vmx_client.

Value

A tibble of log lines.

vmx_model_build_report *Build-run report status (signed HTML report URL when ready)*

Description

Build-run report status (signed HTML report URL when ready)

Usage

```
vmx_model_build_report(run, client = vmx_client())
```

Arguments

run	A build-run id or object.
client	A vmx_client.

Value

A list with status and, when ready, url.

vmx_model_build_report_create *Request build-run report generation*

Description

POST /model-build-runs/{run_id}/report queues HTML report generation.

Usage

```
vmx_model_build_report_create(
  run,
  subject_plot_mode = c("all", "none"),
  client = vmx_client()
)
```

Arguments

run	A build-run id or object.
subject_plot_mode	One of "all" or "none".
client	A vmx_client.

Value

A list with report status.

vmx_model_build_results *Build-run results*

Description

Build-run results

Usage

```
vmx_model_build_results(run, client = vmx_client())
```

Arguments

run	A build-run id or object.
client	A vmx_client.

Value

A list (fits summary, modeling population, PK structure selection).

vmx_model_build_runs *List model build runs*

Description

List model build runs

Usage

```
vmx_model_build_runs(
  data_version = NULL,
  study = NULL,
  treatment = NULL,
  status = NULL,
  client = vmx_client()
)
```

Arguments

data_version, study, treatment, status	Optional filters.
client	A vmx_client.

Value

A tibble.

vmx_model_build_status *Build-run status*

Description

Build-run status

Usage

```
vmx_model_build_status(run, client = vmx_client())
```

Arguments

run	A build-run id (run_...) or object.
client	A vmx_client.

Value

A vmx_model_build_run.

vmx_model_catalog *Model catalog*

Description

GET /model-catalog returns categories of models; this flattens them into one tibble with a category column.

Usage

```
vmx_model_catalog(data_version = NULL, client = vmx_client())
```

Arguments

data_version	Optional data-version to tailor the catalog to.
client	A vmx_client.

Value

A tibble.

vmx_model_data *Fetch model-ready tidy tables for a data version*

Description

Returns a vmx_model_data bundle with $subjects, $pk, $pd (each a tibble, or NULL when the DataVersion has no such prepared table), and $meta (units, time bases, PD-marker manifest, subject count) read from the DataVersion. Only domains flagged in the DV's table_availability are fetched, so absent optional tables don't 404.

Usage

```
vmx_model_data(dv, client = vmx_client())
```

Arguments

dv	A data-version id or vmx_data_version.
client	A vmx_client.

Value

A vmx_model_data object.

vmx_model_describe *Describe a model*

Description

Describe a model

Usage

```
vmx_model_describe(model_name, client = vmx_client())
```

Arguments

model_name	Catalog model name.
client	A vmx_client.

Value

A named list.

vmx_model_fit *Fetch one model fit's details*

Description

Fetch one model fit's details

Usage

```
vmx_model_fit(id, client = vmx_client())
```

Arguments

id	A fit id (mf_...) or vmx_model_fit.
client	A vmx_client.

Value

A vmx_model_fit (metadata / model / inference).

vmx_model_fit_postprocessor_status *Model-fit postprocessor status*

Description

Model-fit postprocessor status

Usage

```
vmx_model_fit_postprocessor_status(fit, client = vmx_client())
```

Arguments

fit	A fit id or vmx_model_fit.
client	A vmx_client.

Value

A list with postprocessor status.

vmx_model_fits *List model fits*

Description

List model fits

Usage

```
vmx_model_fits(
  run = NULL,
  data_version = NULL,
  model_type = NULL,
  marker_name = NULL,
  status = NULL,
  client = vmx_client()
)
```

Arguments

run, data_version, model_type, marker_name, status	Optional filters.
client	A vmx_client.

Value

A tibble.

vmx_modeling_options *Preview modeling options for a data version*

Description

Preview modeling options for a data version

Usage

```
vmx_modeling_options(
  data_version,
  time_basis,
  pd_marker = NULL,
  covariate = NULL,
  client = vmx_client()
)
```

Arguments

data_version	A data-version id or vmx_data_version.
time_basis	Time basis.
pd_marker	Optional PD marker gen_uuid(s) (character vector).
covariate	Optional covariate name(s).
client	A vmx_client.

Value

A list (the selection preview).

---

vmx_nca *Run an NCA analysis*

Description

Creates the analysis (POST /nca-analyses) and, by default, blocks until it settles. time_basis is one of "observed", "nominal", or "nominal_from_observed_dose" (validated server-side against the DataVersion's available bases).

Usage

```
vmx_nca(
  data_version,
  time_basis,
  idempotency_key = NULL,
  retried_from = NULL,
  wait = TRUE,
  ...,
  client = vmx_client()
)
```

Arguments

data_version	A data-version id (dv_...) or vmx_data_version.
time_basis	The time basis to compute on.
idempotency_key, retried_from	Optional create fields.
wait	If TRUE (default), block until the analysis is terminal.
...	Polling controls forwarded to vmx_wait() when wait = TRUE (e.g. timeout, interval, progress).
client	A vmx_client.

Value

A vmx_nca_analysis.

vmx_nca_analyses *List NCA analyses*

Description

List NCA analyses

Usage

```
vmx_nca_analyses(
  data_version = NULL,
  study = NULL,
  treatment = NULL,
  status = NULL,
  time_basis = NULL,
  client = vmx_client()
)
```

Arguments

data_version	Optional data-version (dv_...) filter.
study	Optional study (std_...) filter.
treatment	Optional treatment (tmt_...) filter.
status	Optional status filter (queued/running/completed/ degraded/failed).
time_basis	Optional time-basis filter.
client	A vmx_client.

Value

A tibble, one row per analysis.

vmx_nca_get *Fetch one NCA analysis*

Description

Fetch one NCA analysis

Usage

```
vmx_nca_get(id, client = vmx_client())
```

Arguments

id	An NCA id (nca_...) or vmx_nca_analysis.
client	A vmx_client.

Value

A vmx_nca_analysis.

---

vmx_nca_result *NCA result table (PK parameters, one row per subject)*

Description

Reshapes the point_estimates payload (metric -> per-subject values, parallel to the subject arrays) into a tidy tibble: subject_id, gen_subject_uuid, and one column per PK quantity. The quantity metadata (display names, units, explanations) is attached as the "quantities" attribute.

Usage

```
vmx_nca_result(nca, client = vmx_client())
```

Arguments

nca	An NCA id or vmx_nca_analysis.
client	A vmx_client.

Value

A tibble.

---

vmx_nlmixr_data *NONMEM-layout data.frame for nlmixr2 / rxode2*

Description

Not yet implemented. Assembling the NONMEM/nlmixr2 layout (ID/TIME/DV/AMT/EVID/CMT/MDV/RATE/II/ADDL/
+ covariates) from the pk and dosing domain tables requires the DataVersion column/manifest contract to be pinned and validated against real data; see the package NEWS. Use vmx_pk() / vmx_data_version_table() for the tidy tables today.

Usage

```
vmx_nlmixr_data(dv, analyte = NULL, client = vmx_client())
```

Arguments

dv	A data-version id or vmx_data_version.
analyte	Analyte to assemble.
client	A vmx_client.

Value

Not yet implemented.

---

vmx_pd *PD observations table*

Description

PD observations table

Usage

```
vmx_pd(dv, client = vmx_client())
```

Arguments

dv	A data-version id or vmx_data_version.
client	A vmx_client.

Value

A tibble.

---

vmx_pk *PK observations + events table*

Description

PK observations + events table

Usage

```
vmx_pk(dv, client = vmx_client())
```

Arguments

dv	A data-version id or vmx_data_version.
client	A vmx_client.

Value

A tibble.

vmx_prep_answer *Answer prep questions and resume formatting*

Description

Answer prep questions and resume formatting

Usage

```
vmx_prep_answer(dataset, answers, client = vmx_client())
```

Arguments

dataset	A dataset id or vmx_dataset.
answers	A named list mapping each prompt field to its answer value.
client	A vmx_client.

vmx_prep_questions *Questions raised by prep (when awaiting input)*

Description

Questions raised by prep (when awaiting input)

Usage

```
vmx_prep_questions(dataset, client = vmx_client())
```

Arguments

dataset	A dataset id or vmx_dataset.
client	A vmx_client.

Value

A tibble of pending questions.

vmx_prep_status *Prep status for a dataset*

Description

Calls `GET /datasets/{ds_id}/prep-status`.

Usage

```
vmx_prep_status(dataset, client = vmx_client())
```

Arguments

dataset	A dataset id (`ds_...`) or `vmx_dataset`.
client	A `vmx_client`.

Value

A `vmx_prep_status` (`status`, and `data_version_id` once settled).

vmx_sim_cancel *Cancel a simulation job*

Description

Cancel a simulation job

Usage

```
vmx_sim_cancel(job, client = vmx_client())
```

Arguments

job	A job id or `vmx_simulation_job`.
client	A `vmx_client`.

Value

A `vmx_simulation_job`.

vmx_sim_existing_subject *Simulate existing (observed) subjects*

Description

POST /model-fits/{mf_id}/existing-subject-simulation-jobs.

Usage

```
vmx_sim_existing_subject(
  fit,
  dosing_input,
  subjects,
  idempotency_key = NULL,
  retried_from = NULL,
  min_timepoints = NULL,
  wait = FALSE,
  ...,
  client = vmx_client()
)
```

Arguments

fit	A fit id or vmx_model_fit.
dosing_input	A dosing-input id or vmx_dosing_input.
subjects	Subjects to simulate: a data.frame/tibble with gen_subject_uuid + subject_name columns, or a list of such records.
idempotency_key, retried_from	Optional create fields.
min_timepoints	Optional minimum number of simulated timepoints.
wait	If TRUE, block until the job settles.
...	Polling controls forwarded to vmx_wait().
client	A vmx_client.

Value

A vmx_simulation_job.

vmx_sim_existing_subject_from_text
Simulate existing subjects from dosing text

Description

POST /model-fits/{mf_id}/existing-subject-simulation-jobs/from-text.

Usage

```
vmx_sim_existing_subject_from_text(
  fit,
  dosing_text,
  subjects,
  idempotency_key = NULL,
  retried_from = NULL,
  min_timepoints = NULL,
  wait = FALSE,
  ...,
  client = vmx_client()
)
```

Arguments

fit	A fit id or vmx_model_fit.
dosing_text	The dosing regimen text.
subjects	Subjects to simulate: a data.frame/tibble with gen_subject_uuid + subject_name columns, or a list of such records.
idempotency_key, retried_from	Optional create fields.
min_timepoints	Optional minimum number of simulated timepoints.
wait	If TRUE, block until the job settles.
...	Polling controls forwarded to vmx_wait().
client	A vmx_client.

Value

A vmx_simulation_job.

vmx_sim_hypothetical_subject *Simulate hypothetical subjects*

Description

`POST /model-fits/{mf_id}/hypothetical-subject-simulation-jobs`.

Usage

```
vmx_sim_hypothetical_subject(
  fit,
  dosing_input,
  subjects,
  idempotency_key = NULL,
  retried_from = NULL,
  min_timepoints = NULL,
  wait = FALSE,
  ...,
  client = vmx_client()
)
```

Arguments

`fit`	A fit id or `vmx_model_fit`.
`dosing_input`	A dosing-input id or `vmx_dosing_input`.
`subjects`	A data.frame/tibble with a `subject_name` column plus one column per covariate, or a list of `{subject_name, covariates}` records.
`idempotency_key`, `retried_from`	Optional create fields.
`min_timepoints`	Optional minimum number of simulated timepoints.
`wait`	If `TRUE`, block until the job settles.
`...`	Polling controls forwarded to `vmx_wait()`.
`client`	A `vmx_client`.

Value

A `vmx_simulation_job`.

`vmx_sim_hypothetical_subject_from_text` *Simulate hypothetical subjects from dosing text*

Description

`POST /model-fits/{mf_id}/hypothetical-subject-simulation-jobs/from-text`.

Usage

```
vmx_sim_hypothetical_subject_from_text(
  fit,
  dosing_text,
  subjects,
  idempotency_key = NULL,
  retried_from = NULL,
  min_timepoints = NULL,
  wait = FALSE,
  ...,
  client = vmx_client()
)
```

Arguments

`fit`	A fit id or `vmx_model_fit`.
`dosing_text`	The dosing regimen text.
`subjects`	A data.frame/tibble with a `subject_name` column plus one column per covariate, or a list of `{subject_name, covariates}` records.
`idempotency_key, retried_from`	Optional create fields.
`min_timepoints`	Optional minimum number of simulated timepoints.
`wait`	If `TRUE`, block until the job settles.
`...`	Polling controls forwarded to `vmx_wait()`.
`client`	A `vmx_client`.

Value

A `vmx_simulation_job`.

vmx_sim_jobs *List simulation jobs for a model fit*

Description

List simulation jobs for a model fit

Usage

```
vmx_sim_jobs(fit, client = vmx_client())
```

Arguments

fit	A fit id or vmx_model_fit.
client	A vmx_client.

Value

A tibble.

vmx_sim_population *Simulate a population scenario*

Description

POST /model-fits/{mf_id}/population-simulation-jobs.

Usage

```
vmx_sim_population(
  fit,
  dosing_input,
  scenario_name,
  idempotency_key = NULL,
  retried_from = NULL,
  min_timepoints = NULL,
  wait = FALSE,
  ...,
  client = vmx_client()
)
```

Arguments

fit	A fit id or vmx_model_fit.
dosing_input	A dosing-input id or vmx_dosing_input.
scenario_name	The population scenario name.
idempotency_key, retried_from	Optional create fields.
min_timepoints	Optional minimum number of simulated timepoints.
wait	If TRUE, block until the job settles.
...	Polling controls forwarded to vmx_wait().
client	A vmx_client.

Value

A vmx_simulation_job.

---

vmx_sim_population_from_text

Simulate a population scenario from dosing text

Description

POST /model-fits/{mf_id}/population-simulation-jobs/from-text.

Usage

```
vmx_sim_population_from_text(
  fit,
  dosing_text,
  scenario_name,
  idempotency_key = NULL,
  retried_from = NULL,
  min_timepoints = NULL,
  wait = FALSE,
  ...,
  client = vmx_client()
)
```

Arguments

fit	A fit id or vmx_model_fit.
dosing_text	The dosing regimen text.
scenario_name	The population scenario name.
idempotency_key, retried_from	Optional create fields.

min_timepoints	Optional minimum number of simulated timepoints.
wait	If TRUE, block until the job settles.
...	Polling controls forwarded to vmx_wait().
client	A vmx_client.

Value

A vmx_simulation_job.

---

vmx_sim_result *Simulation result*

Description

GET /simulation-jobs/{id}/result. Returns the parsed result payload (subject/time series with prediction bands). Tibble reshaping is deferred pending confirmation of the artifact shape; see the package NEWS.

Usage

```
vmx_sim_result(job, grouping_variable = NULL, client = vmx_client())
```

Arguments

job	A job id or vmx_simulation_job.
grouping_variable	Optional server-side grouping.
client	A vmx_client.

Value

A list (the parsed result).

---

vmx_sim_status *Simulation job status*

Description

Simulation job status

Usage

```
vmx_sim_status(job, client = vmx_client())
```

Arguments

job	A job id (simjob_...) or vmx_simulation_job.
client	A vmx_client.

Value

A vmx_simulation_job.

---

vmx_studies *List studies for a treatment*

Description

List studies for a treatment

Usage

```
vmx_studies(treatment = NULL, status = NULL, client = vmx_client())
```

Arguments

treatment	A treatment id (tmt_...) or vmx_treatment; NULL lists across all treatments.
status	Optional status filter.
client	A vmx_client.

Value

A tibble, one row per study.

---

vmx_study *Fetch one study*

Description

Fetch one study

Usage

```
vmx_study(id, client = vmx_client())
```

Arguments

id	A study id (std_...) or vmx_study object.
client	A vmx_client.

Value

A vmx_study.

vmx_study_create *Create a study*

Description

Create a study

Usage

```
vmx_study_create(
  treatment,
  name,
  study_type = "clinical",
  phase = NULL,
  ...,
  client = vmx_client()
)
```

Arguments

treatment	A treatment id or vmx_treatment.
name	Study name.
study_type	Study type; defaults to "clinical".
phase	Optional clinical phase.
...	Additional fields (description, route_of_administration, pd_markers).
client	A vmx_client.

Value

A vmx_study.

vmx_study_update *Update a study*

Description

Only the fields you pass are changed (the server applies exclude_unset).

Usage

```
vmx_study_update(id, ..., client = vmx_client())
```

Arguments

id	A study id or vmx_study.
...	Fields to update.
client	A vmx_client.

Value

A vmx_study.

---

vmx_subjects *Subjects table (one row per subject)*

Description

Subjects table (one row per subject)

Usage

```
vmx_subjects(dv, client = vmx_client())
```

Arguments

dv	A data-version id or vmx_data_version.
client	A vmx_client.

Value

A tibble.

---

vmx_torsten_data *Ragged-array data list for Stan / Torsten*

Description

Not yet implemented. The per-subject start[i]/end[i] index ranges and iObs observation index must be derived and verified against real data before shipping (this is the error-prone derivation the design flags); see the package NEWS.

Usage

```
vmx_torsten_data(dv, analyte = NULL, client = vmx_client())
```

Arguments

dv	A data-version id or vmx_data_version.
analyte	Analyte to assemble.
client	A vmx_client.

Value

Not yet implemented.

---

vmx_treatment *Fetch one treatment*

Description

Fetch one treatment

Usage

```
vmx_treatment(id, client = vmx_client())
```

Arguments

id	A treatment id (tmt_...) or vmx_treatment object.
client	A vmx_client.

Value

A vmx_treatment.

---

vmx_treatment_create *Create a treatment*

Description

Create a treatment

Usage

```
vmx_treatment_create(
  name,
  indication = NULL,
  description = NULL,
  client = vmx_client()
)
```

Arguments

name	Treatment name.
indication	Optional indication.
description	Optional free-text description.
client	A vmx_client.

Value

A vmx_treatment.

---

vmx_treatment_update *Update a treatment*

Description

Only the fields you pass are changed (the server applies exclude_unset).

Usage

```
vmx_treatment_update(id, ..., client = vmx_client())
```

Arguments

id	A treatment id or vmx_treatment.
...	Fields to update (name, indication, description, status).
client	A vmx_client.

Value

A vmx_treatment.

---

vmx_treatments *List treatments*

Description

List treatments

Usage

```
vmx_treatments(status = NULL, client = vmx_client())
```

Arguments

status	Optional status filter.
client	A vmx_client.

Value

A tibble, one row per treatment.

---

vmx_upload *Upload files to a study*

Description

Streamed multipart upload. With wait = TRUE, blocks through the prep pipeline (see vmx_wait()).

Usage

```
vmx_upload(
  study,
  files,
  mode = c("initial", "incremental", "replacement"),
  treatment = NULL,
  config = NULL,
  wait = FALSE,
  client = vmx_client()
)
```

Arguments

study	A study id or vmx_study.
files	Character vector of local file paths.
mode	One of "initial" (auto-formats a default DataVersion), "incremental", or "replacement".
treatment	Optional treatment; inferred from study when possible.
config	Optional gecodata v2 project.yaml path (warm-start).
wait	If TRUE, block until prep settles.
client	A vmx_client.

Value

A vmx_dataset (status "uploaded").

vmx_upload_ignore *Ignore an upload within a dataset*

Description

`POST /datasets/{ds_id}/ignore-upload`. Soft-ignores one delivery.

Usage

```
vmx_upload_ignore(dataset, upload, client = vmx_client())
```

Arguments

`dataset`	A dataset id or `vmx_dataset`.
`upload`	The upload id (`upl_...`) to ignore.
`client`	A `vmx_client`.

Value

The updated `vmx_prep_status`.

vmx_upload_unignore *Reverse a prior upload ignore*

Description

`POST /datasets/{ds_id}/unignore-upload`.

Usage

```
vmx_upload_unignore(dataset, upload, client = vmx_client())
```

Arguments

`dataset`	A dataset id or `vmx_dataset`.
`upload`	The upload id (`upl_...`) to ignore.
`client`	A `vmx_client`.

Value

The updated `vmx_prep_status`.

vmx_wait *Block until an async handle reaches a terminal state*

Description

An S3 generic dispatching on the handle type. Each method has a sensible default terminal state. Terminal-but-unsuccessful states raise a classed error so scripts fail loudly rather than hang.

Usage

```
vmx_wait(
  x,
  until = NULL,
  timeout = 900,
  interval = 5,
  progress = interactive(),
  client = vmx_client(),
  ...
)
```

Arguments

x	A pollable handle: a dataset / prep-status or an NCA analysis (more types as the API surface lands: model-build-run, simulation-job).
until	Target terminal state(s); a sensible default per type when NULL.
timeout	Timeout in seconds.
interval	Poll interval in seconds (exponential backoff up to 30s).
progress	Show a progress message each poll; defaults to interactive().
client	A vmx_client.
...	Passed to methods.

Value

The updated object, or a vmx_timeout_error.

vmx_whoami *Confirm the configured PAT and report identity*

Description

Calls GET /me.

Usage

```
vmx_whoami(client = vmx_client())
```

Arguments

client	A vmx_client.

Value

A vmx_me object: user_id, email, name, workspace_id, roles, and counts.

Index