Package ‘rchroma’

February 13, 2025

Type Package

Title A Client for 'ChromaDB'

Version 0.1.0

Description Client for 'ChromaDB', a vector database for storing and querying embeddings. This package provides a convenient interface to interact with the REST API of 'ChromaDB' <https://docs.trychroma.com>.

License MIT + file LICENSE

URL https://github.com/cynkra/rchroma

BugReports https://github.com/cynkra/rchroma/issues

Language en-US

Encoding UTF-8

RoxygenNote 7.3.2

Imports httr2 (>= 1.0.0)

Suggests testthat (>= 3.0.0), knitr, rmarkdown, covr, spelling

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author David Schoch [aut, cre] (<https://orcid.org/0000-0003-2952-4812>),
Christoph Sax [aut],
cynkra LLC [cph]

Maintainer David Schoch <david@schochastics.net>

Repository CRAN

Date/Publication 2025-02-13 11:30:09 UTC

Contents

`add_documents` *Add Documents to a Collection*

Description

Add Documents to a Collection

Usage

```
add_documents(
  client,
  collection_name,
  documents,
  ids,
  metadatas = NULL,
  embeddings = NULL,
  uris = NULL,
  tenant = "default_tenant",
  database = "default_database"
)
```

Arguments

`client`	A ChromaDB client object
`collection_name`	Name of the collection
`documents`	List of documents to add

ids	Vector of unique IDs for the documents (required)
metadatas	List of metadata for each document (optional)
embeddings	Optional pre-computed embeddings
uris	Optional vector of URIs associated with the documents
tenant	The tenant name (default: "default")
database	The database name (default: "default")

Value

TRUE on success

chroma_connect *ChromaDB Client*

Description

Create a new ChromaDB client connection

Usage

```
chroma_connect(
  host = "http://localhost",
  port = 8000L,
  api_path = "/api/v2",
  verify = TRUE
)
```

Arguments

host	The host URL of the ChromaDB server. Default is "http://localhost".
port	The port number of the ChromaDB server. Default is 8000.
api_path	The API path. Default is "/api/v2".
verify	Whether to verify the connection. Default is TRUE.

Value

A ChromaDB client object

count_collections *Count Collections in a Database*

Description

Count Collections in a Database

Usage

```
count_collections(
  client,
  tenant = "default_tenant",
  database = "default_database"
)
```

Arguments

`client`	A ChromaDB client object
`tenant`	The tenant name (default: "default")
`database`	The database name (default: "default")

Value

Number of collections in the database

create_collection *Create a Collection in ChromaDB*

Description

Create a Collection in ChromaDB

Usage

```
create_collection(
  client,
  name,
  metadata = NULL,
  configuration = NULL,
  tenant = "default_tenant",
  database = "default_database",
  get_or_create = FALSE
)
```

Arguments

`client`	A ChromaDB client object
`name`	The name of the collection
`metadata`	Optional metadata for the collection
`configuration`	Optional configuration for the collection. For HNSW configuration, use a list with `hnsw_space` (e.g., "cosine", "l2", "ip").
`tenant`	The tenant name (default: "default")
`database`	The database name (default: "default")
`get_or_create`	Whether to get the collection if it exists (default: FALSE)

Value

A collection object

`create_database` *Create a Database*

Description

Create a Database

Usage

```
create_database(client, name, tenant = "default_tenant")
```

Arguments

`client`	A ChromaDB client object
`name`	The name of the database
`tenant`	The tenant name

Value

NULL invisibly on success

create_tenant *Create a Tenant*

Description

Create a Tenant

Usage

```
create_tenant(client, name)
```

Arguments

`client`	A ChromaDB client object
`name`	The name of the tenant

Value

A tenant object containing the tenant details

delete_collection *Delete a Collection*

Description

Delete a Collection

Usage

```
delete_collection(
  client,
  name,
  tenant = "default_tenant",
  database = "default_database"
)
```

Arguments

`client`	A ChromaDB client object
`name`	The name of the collection
`tenant`	The tenant name (default: "default")
`database`	The database name (default: "default")

Value

Invisible NULL on success

delete_documents *Delete Documents from a Collection*

Description

Delete Documents from a Collection

Usage

```
delete_documents(
  client,
  collection_name,
  ids = NULL,
  where = NULL,
  tenant = "default_tenant",
  database = "default_database"
)
```

Arguments

`client`	A ChromaDB client object
`collection_name`	Name of the collection
`ids`	Vector of document IDs to delete
`where`	Optional filtering conditions
`tenant`	The tenant name (default: "default")
`database`	The database name (default: "default")

Value

NULL invisibly on success

get_auth_identity *Get Authentication Identity*

Description

Get Authentication Identity

Usage

```
get_auth_identity(client)
```

Arguments

`client`	A ChromaDB client object

Value

Authentication identity information

`get_collection` *Get a Collection*

Description

Get a Collection

Usage

```
get_collection(
  client,
  name,
  tenant = "default_tenant",
  database = "default_database"
)
```

Arguments

`client`	A ChromaDB client object
`name`	The name of the collection
`tenant`	The tenant name (default: "default")
`database`	The database name (default: "default")

Value

A collection object

`get_database` *Get a Database*

Description

Get a Database

Usage

```
get_database(client, name, tenant = "default_tenant")
```

Arguments

`client`	A ChromaDB client object
`name`	The name of the database
`tenant`	The tenant name

Value

NULL invisibly on success

`get_tenant` *Get a Tenant*

Description

Get a Tenant

Usage

```
get_tenant(client, name)
```

Arguments

`client`	A ChromaDB client object
`name`	The name of the tenant

Value

A tenant object containing the tenant details

`heartbeat` *Check ChromaDB Server Heartbeat*

Description

Check ChromaDB Server Heartbeat

Usage

```
heartbeat(client)
```

Arguments

`client`	A ChromaDB client object

Value

Server heartbeat response as a numeric value

`list_collections` *List Collections in a Database*

Description

List Collections in a Database

Usage

```
list_collections(
  client,
  tenant = "default_tenant",
  database = "default_database",
  limit = NULL,
  offset = NULL
)
```

Arguments

`client`	A ChromaDB client object
`tenant`	The tenant name (default: "default")
`database`	The database name (default: "default")
`limit`	Maximum number of collections to return (optional)
`offset`	Number of collections to skip (optional)

Value

List of collections

`pre_flight_checks` *Get ChromaDB Server Information*

Description

Returns server capabilities and settings.

Usage

```
pre_flight_checks(client)
```

Arguments

`client`	A ChromaDB client object

Value

List containing server information

`query` *Query Documents in a Collection*

Description

Query Documents in a Collection

Usage

```
query(
  client,
  collection_name,
  query_embeddings,
  n_results = 10L,
  where = NULL,
  where_document = NULL,
  include = c("documents", "metadatas", "distances"),
  tenant = "default_tenant",
  database = "default_database"
)
```

Arguments

client A ChromaDB client object

collection_name
Name of the collection

query_embeddings
List of query embeddings (must be a list of numeric vectors)

n_results Number of results to return per query (default: 10)

where Optional filtering conditions

where_document Optional document-based filtering conditions

include Optional vector of what to include in results. Possible values: "documents", "embeddings", "metadatas", "distances", "uris", "data" (default: c("documents", "metadatas", "distances"))

tenant The tenant name (default: "default")

database The database name (default: "default")

Details

Note that ChromaDB's API only accepts embeddings for queries. If you want to query using text, you need to first convert your text to embeddings using an embedding model (e.g., using OpenAI's API, HuggingFace's API, or a local model).

Example:

```
# First convert text to embeddings using your preferred method
text_embedding <- your_embedding_function("your search text")
# Then query using the embedding
result <- query(client, "my_collection",
               query_embeddings = list(text_embedding))
```

Value

A list containing the query results. Each element (documents, metadatas, distances) is a nested list, so use double brackets [[]] to access individual elements.

reset *Reset ChromaDB*

Description

This function resets the entire ChromaDB instance. Use with caution as this will delete all data. Note: This function requires setting ALLOW_RESET=TRUE in the environment variables or allow_reset=True in the ChromaDB Settings.

Usage

```
reset(client)
```

Arguments

client	A ChromaDB client object

Value

TRUE on success

`update_collection` *Update a Collection*

Description

Update a Collection

Usage

```
update_collection(
  client,
  name,
  new_name = NULL,
  new_metadata = NULL,
  tenant = "default_tenant",
  database = "default_database"
)
```

Arguments

client	A ChromaDB client object
name	The name of the collection
new_name	Optional new name for the collection
new_metadata	Optional new metadata for the collection
tenant	The tenant name (default: "default")
database	The database name (default: "default")

Value

NULL on success (invisibly)

update_documents *Update Documents in a Collection*

Description

Update Documents in a Collection

Usage

```
update_documents(
  client,
  collection_name,
  ids,
  documents = NULL,
  metadatas = NULL,
  embeddings = NULL,
  tenant = "default_tenant",
  database = "default_database"
)
```

Arguments

`client`	A ChromaDB client object
`collection_name`	Name of the collection
`ids`	Vector of document IDs to update
`documents`	List of new document contents
`metadatas`	List of new metadata
`embeddings`	Optional new pre-computed embeddings
`tenant`	The tenant name (default: "default")
`database`	The database name (default: "default")

Value

NULL invisibly on success

`upsert_documents` *Upsert Documents to a Collection*

Description

Upsert Documents to a Collection

Usage

```
upsert_documents(
  client,
  collection_name,
  documents,
  metadatas = NULL,
  ids = NULL,
  embeddings = NULL,
  uris = NULL
)
```

Arguments

`client`	A ChromaDB client object
`collection_name`	Name of the collection
`documents`	List of documents to upsert
`metadatas`	List of metadata for each document
`ids`	Vector of unique IDs for the documents
`embeddings`	Optional pre-computed embeddings
`uris`	Optional vector of URIs associated with the documents

Value

Response from the API

`version` *Get ChromaDB Server Version*

Description

Get ChromaDB Server Version

Usage

```
version(client)
```

Arguments

`client`	A ChromaDB client object

Value

Server version string

Index