

ChromaDB Documentation

ChromaDB is a specialized module designed to facilitate the storage and retrieval of documents using the ChromaDB system. It offers functionalities for adding documents to a local ChromaDB collection and querying this collection based on provided query texts. This module integrates with the ChromaDB client to create and manage collections, leveraging various configurations for optimizing the storage and retrieval processes.

Parameters

Parameter	Type	Default	Description
`metric`	`str`	`"cosine"`	The similarity metric to use for the collection.
`output_dir`	`str`	`"swarms"`	The name of the collection to store the results in.
`limit_tokens`	`Optional[int]`	`1000`	The maximum number of tokens to use for the query.
`n_results`	`int`	`1`	The number of results to retrieve.
`docs_folder`	`Optional[str]`	`None`	The folder containing documents to be added to the collection.
`verbose`	`bool`	`False`	Flag to enable verbose logging for debugging.
`*args`	`tuple`	`()`	Additional positional arguments.
`**kwargs`	`dict`	`{}`	Additional keyword arguments.

Methods

Method	Description
----- -----	
<code>`__init__`</code>	Initializes the ChromaDB instance with specified parameters.
<code>`add`</code>	Adds a document to the ChromaDB collection.
<code>`query`</code>	Queries documents from the ChromaDB collection based on the query text.
<code>`traverse_directory`</code>	Traverses the specified directory to add documents to the collection.

Usage

```
```python
```

```
from swarms_memory import ChromaDB
```

```
chromadb = ChromaDB(
 metric="cosine",
 output_dir="results",
 limit_tokens=1000,
 n_results=2,
 docs_folder="path/to/docs",
 verbose=True,
)
```
```

Adding Documents

The ``add`` method allows you to add a document to the ChromaDB collection. It generates a unique

ID for each document and adds it to the collection.

Parameters

| Parameter | Type | Default | Description | |
|-----------|-------|---------|---|--|
| document | str | - | The document to be added to the collection. | |
| *args | tuple | () | Additional positional arguments. | |
| **kwargs | dict | {} | Additional keyword arguments. | |

Returns

| Type | Description | |
|------|-------------------------------|--|
| str | The ID of the added document. | |

Example

```
```python
task = "example_task"
result = "example_result"
result_id = chromadb.add(document="This is a sample document.")
print(f"Document ID: {result_id}")
...
```
```

Querying Documents

The ``query`` method allows you to retrieve documents from the ChromaDB collection based on the provided query text.

Parameters

| Parameter | Type | Default | Description | |
|---------------------------|----------------------|-------------------|----------------------------------|--|
| ----- ----- ----- ----- | | | | |
| <code>`query_text`</code> | <code>`str`</code> | - | The query string to search for. | |
| <code>`*args`</code> | <code>`tuple`</code> | <code>`()`</code> | Additional positional arguments. | |
| <code>`**kwargs`</code> | <code>`dict`</code> | <code>`{}`</code> | Additional keyword arguments. | |

Returns

| Type | Description | |
|--------------------|--------------------------------------|--|
| ----- ----- | | |
| <code>`str`</code> | The retrieved documents as a string. | |

Example

```
```python
query_text = "search term"

results = chromadb.query(query_text=query_text)

print(f"Retrieved Documents: {results}")
```
```

Traversing Directory

The `traverse_directory` method traverses through every file in the specified directory and its subdirectories, adding the contents of each file to the ChromaDB collection.

Example

```
```python
chromadb.traverse_directory()
```
```

Additional Information and Tips

Verbose Logging

Enable the `verbose` flag during initialization to get detailed logs of the operations, which is useful for debugging.

```
```python
chromadb = ChromaDB(verbose=True)
```
```

Handling Large Documents

When dealing with large documents, consider using the `limit_tokens` parameter to restrict the number of tokens processed in a single query.

```
```python
```

```
chromadb = ChromaDB(limit_tokens=500)
```

```
```
```

Optimizing Query Performance

Use the appropriate similarity metric (`metric` parameter) that suits your use case for optimal query performance.

```
```python
```

```
chromadb = ChromaDB(metric="euclidean")
```

```
```
```

References and Resources

- [ChromaDB Documentation](<https://chromadb.io/docs>)
- [Python UUID Module](<https://docs.python.org/3/library/uuid.html>)
- [Python os Module](<https://docs.python.org/3/library/os.html>)
- [Python logging Module](<https://docs.python.org/3/library/logging.html>)
- [dotenv Package](<https://pypi.org/project/python-dotenv/>)

By following this documentation, users can effectively utilize the ChromaDB module for managing document storage and retrieval in their applications.