

Chroma

Chroma is a AI-native open-source vector database focused on developer productivity and happiness. Chroma is licensed under Apache 2.0.

Install Chroma with:

```
pip install chromadb
```

Chroma runs in various modes. See below for examples of each integrated with LangChain.

- in-memory - in a python script or jupyter notebook
- in-memory with persistence - in a script or notebook and save/load to disk
- in a docker container - as a server running your local machine or in the cloud

Like any other database, you can:

- .add
- .get
- .update
- .upsert
- .delete
- .peek
- and .query runs the similarity search.

View full docs at docs. To access these methods directly, you can do `._collection.method()`

Basic Example

In this basic example, we take the most recent State of the Union Address, split it into chunks, embed it using an open-source embedding model, load it into Chroma, and then query it.

```
# import
from langchain.document_loaders import TextLoader
from langchain.embeddings.sentence_transformer import
SentenceTransformerEmbeddings
from langchain.text_splitter import CharacterTextSplitter
from langchain.vectorstores import Chroma
```

```
# load the document and split it into chunks
```

```
loader = TextLoader("../modules/state_of_the_union.txt")
documents = loader.load()
```

```
# split it into chunks
```

```
text_splitter = CharacterTextSplitter(chunk_size=1000,
chunk_overlap=0)
docs = text_splitter.split_documents(documents)
```

```
# create the open-source embedding function
```

```
embedding_function =
SentenceTransformerEmbeddings(model_name="all-MiniLM-
L6-v2")
```

```
# load it into Chroma
```

```
db = Chroma.from_documents(docs, embedding_function)
```

```
# query it
```

```
query = "What did the president say about Ketanji Brown
Jackson"
docs = db.similarity_search(query)
```

```
# print results
```

```
print(docs[0].page_content)
```

```
/Users/jeff/.pyenv/versions/3.10.10/lib/python3.10/site-
packages/tqdm/auto.py:21: TqdmWarning: IProgress not
found. Please update jupyter and ipywidgets. See https://
ipywidgets.readthedocs.io/en/stable/user_install.html
from .autonotebook import tqdm as notebook_tqdm
```

Tonight. I call on the Senate to: Pass the Freedom to Vote Act. Pass the John Lewis Voting Rights Act. And while you're at it, pass the Disclose Act so Americans can know who is funding our elections.

Tonight, I'd like to honor someone who has dedicated his life to serve this country: Justice Stephen Breyer—an Army veteran, Constitutional scholar, and retiring Justice of the

United States Supreme Court. Justice Breyer, thank you for your service.

One of the most serious constitutional responsibilities a President has is nominating someone to serve on the United States Supreme Court.

And I did that 4 days ago, when I nominated Circuit Court of Appeals Judge Ketanji Brown Jackson. One of our nation's top legal minds, who will continue Justice Breyer's legacy of excellence.

Basic Example (including saving to disk)

Extending the previous example, if you want to save to disk, simply initialize the Chroma client and pass the directory where you want the data to be saved to.

Caution: Chroma makes a best-effort to automatically save data to disk, however multiple in-memory clients can stomp each other's work. As a best practice, only have one client per path running at any given time.

`# save to disk`

```
db2 = Chroma.from_documents(docs, embedding_function,  
persist_directory="./chroma_db")  
docs = db2.similarity_search(query)
```

`# load from disk`

```
db3 = Chroma(persist_directory="./chroma_db",  
embedding_function=embedding_function)  
docs = db3.similarity_search(query)  
print(docs[0].page_content)
```

Tonight. I call on the Senate to: Pass the Freedom to Vote Act. Pass the John Lewis Voting Rights Act. And while you're at it, pass the Disclose Act so Americans can know who is funding our elections.

Tonight, I'd like to honor someone who has dedicated his life to serve this country: Justice Stephen Breyer—an Army

veteran, Constitutional scholar, and retiring Justice of the United States Supreme Court. Justice Breyer, thank you for your service.

One of the most serious constitutional responsibilities a President has is nominating someone to serve on the United States Supreme Court.

And I did that 4 days ago, when I nominated Circuit Court of Appeals Judge Ketanji Brown Jackson. One of our nation's top legal minds, who will continue Justice Breyer's legacy of excellence.

Passing a Chroma Client into Langchain

You can also create a Chroma Client and pass it to LangChain. This is particularly useful if you want easier access to the underlying database.

You can also specify the collection name that you want LangChain to use.

```
import chromadb
```

```
persistent_client = chromadb.PersistentClient()
collection =
persistent_client.get_or_create_collection("collection_name"
)
collection.add(ids=["1", "2", "3"], documents=["a", "b", "c"])
```

```
langchain_chroma = Chroma(
    client=persistent_client,
    collection_name="collection_name",
    embedding_function=embedding_function,
)
```

```
print("There are", langchain_chroma._collection.count(), "in
the collection")
```

```
Add of existing embedding ID: 1
Add of existing embedding ID: 2
```

```
Add of existing embedding ID: 3
Add of existing embedding ID: 1
Add of existing embedding ID: 2
Add of existing embedding ID: 3
Add of existing embedding ID: 1
Insert of existing embedding ID: 1
Add of existing embedding ID: 2
Insert of existing embedding ID: 2
Add of existing embedding ID: 3
Insert of existing embedding ID: 3
```

There are 3 in the collection

Basic Example (using the Docker Container)

You can also run the Chroma Server in a Docker container separately, create a Client to connect to it, and then pass that to LangChain.

Chroma has the ability to handle multiple Collections of documents, but the LangChain interface expects one, so we need to specify the collection name. The default collection name used by LangChain is "langchain".

Here is how to clone, build, and run the Docker Image:

```
git clone git@github.com:chroma-core/chroma.git
```

Edit the docker-compose.yml file and add ALLOW_RESET=TRUE under environment

```
...
  command: uvicorn chromadb.app:app --reload --workers 1
--host 0.0.0.0 --port 8000 --log-config log_config.yml
  environment:
    - IS_PERSISTENT=TRUE
    - ALLOW_RESET=TRUE
  ports:
    - 8000:8000
  ...
```

Then run docker-compose up -d --build

```
# create the chroma client
```

```
import uuid
```

```
import chromadb
from chromadb.config import Settings
```

```
client =
chromadb.HttpClient(settings=Settings(allow_reset=True))
client.reset() # resets the database
collection = client.create_collection("my_collection")
for doc in docs:
    collection.add(
        ids=[str(uuid.uuid1())], metadatas=doc.metadata,
documents=doc.page_content
    )
```

```
# tell LangChain to use our client and collection name
```

```
db4 = Chroma(
    client=client,
    collection_name="my_collection",
    embedding_function=embedding_function,
)
```

```
query = "What did the president say about Ketanji Brown
Jackson"
```

```
docs = db4.similarity_search(query)
print(docs[0].page_content)
```

Tonight. I call on the Senate to: Pass the Freedom to Vote Act. Pass the John Lewis Voting Rights Act. And while you're at it, pass the Disclose Act so Americans can know who is funding our elections.

Tonight, I'd like to honor someone who has dedicated his life to serve this country: Justice Stephen Breyer—an Army veteran, Constitutional scholar, and retiring Justice of the United States Supreme Court. Justice Breyer, thank you for your service.

One of the most serious constitutional responsibilities a President has is nominating someone to serve on the United

States Supreme Court.

And I did that 4 days ago, when I nominated Circuit Court of Appeals Judge Ketanji Brown Jackson. One of our nation's top legal minds, who will continue Justice Breyer's legacy of excellence.

Update and Delete

While building toward a real application, you want to go beyond adding data, and also update and delete data. Chroma has users provide ids to simplify the bookkeeping here. ids can be the name of the file, or a combined has like filename_paragraphNumber, etc.

Chroma supports all these operations - though some of them are still being integrated all the way through the LangChain interface. Additional workflow improvements will be added soon.

Here is a basic example showing how to do various operations:

```
# create simple ids
ids = [str(i) for i in range(1, len(docs) + 1)]
```

```
# add data
example_db = Chroma.from_documents(docs,
embedding_function, ids=ids)
docs = example_db.similarity_search(query)
print(docs[0].metadata)
```

```
# update the metadata for a document
docs[0].metadata = {
    "source": "../modules/state_of_the_union.txt",
    "new_value": "hello world",
}
example_db.update_document(ids[0], docs[0])
print(example_db._collection.get(ids=[ids[0]]))
```

```
# delete the last document
print("count before", example_db._collection.count())
```

```
example_db._collection.delete(ids=[ids[-1]])  
print("count after", example_db._collection.count())
```

```
{'source': '.././../state_of_the_union.txt'}  
{'ids': ['1'], 'embeddings': None, 'metadatas': [{'new_value':  
'hello world', 'source': '.././../state_of_the_union.txt'}],  
'documents': ['Tonight. I call on the Senate to: Pass the  
Freedom to Vote Act. Pass the John Lewis Voting Rights Act.  
And while you’re at it, pass the Disclose Act so Americans can  
know who is funding our elections. \n\nTonight, I’d like to  
honor someone who has dedicated his life to serve this  
country: Justice Stephen Breyer—an Army veteran,  
Constitutional scholar, and retiring Justice of the United  
States Supreme Court. Justice Breyer, thank you for your  
service. \n\nOne of the most serious constitutional  
responsibilities a President has is nominating someone to  
serve on the United States Supreme Court. \n\nAnd I did that  
4 days ago, when I nominated Circuit Court of Appeals Judge  
Ketanji Brown Jackson. One of our nation’s top legal minds,  
who will continue Justice Breyer’s legacy of excellence.']}  
count before 46  
count after 45
```

Use OpenAI Embeddings

Many people like to use OpenAI Embeddings, here is how to set that up.

get a token: <https://platform.openai.com/account/api-keys>

```
from getpass import getpass
```

```
from langchain.embeddings.openai import  
OpenAIEmbeddings
```

```
OPENAI_API_KEY = getpass()
```

```
import os
```

```
os.environ["OPENAI_API_KEY"] = OPENAI_API_KEY
```



```
embeddings = OpenAIEmbeddings()
new_client = chromadb.EphemeralClient()
openai_lc_client = Chroma.from_documents(
    docs, embeddings, client=new_client,
    collection_name="openai_collection"
)
```

```
query = "What did the president say about Ketanji Brown Jackson"
docs = openai_lc_client.similarity_search(query)
print(docs[0].page_content)
```

Tonight. I call on the Senate to: Pass the Freedom to Vote Act. Pass the John Lewis Voting Rights Act. And while you're at it, pass the Disclose Act so Americans can know who is funding our elections.

Tonight, I'd like to honor someone who has dedicated his life to serve this country: Justice Stephen Breyer—an Army veteran, Constitutional scholar, and retiring Justice of the United States Supreme Court. Justice Breyer, thank you for your service.

One of the most serious constitutional responsibilities a President has is nominating someone to serve on the United States Supreme Court.

And I did that 4 days ago, when I nominated Circuit Court of Appeals Judge Ketanji Brown Jackson. One of our nation's top legal minds, who will continue Justice Breyer's legacy of excellence.

Other Information

Similarity search with score

The returned distance score is cosine distance. Therefore, a lower score is better.

```
docs = db.similarity_search_with_score(query)
```

```
docs[0]
```

```
(Document(page_content='Tonight. I call on the Senate to: Pass the Freedom to Vote Act. Pass the John Lewis Voting Rights Act. And while you're at it, pass the Disclose Act so Americans can know who is funding our elections.\n\nTonight, I'd like to honor someone who has dedicated his life to serve this country: Justice Stephen Breyer—an Army veteran, Constitutional scholar, and retiring Justice of the United States Supreme Court. Justice Breyer, thank you for your service. \n\nOne of the most serious constitutional responsibilities a President has is nominating someone to serve on the United States Supreme Court. \n\nAnd I did that 4 days ago, when I nominated Circuit Court of Appeals Judge Ketanji Brown Jackson. One of our nation's top legal minds, who will continue Justice Breyer's legacy of excellence.', metadata={'source': '.././../state_of_the_union.txt'}), 1.1972057819366455)
```

Retriever options

This section goes over different options for how to use Chroma as a retriever.

MMR

In addition to using similarity search in the retriever object, you can also use mmr.

```
retriever = db.as_retriever(search_type="mmr")
```

```
retriever.get_relevant_documents(query)[0]
```

```
Document(page_content='Tonight. I call on the Senate to: Pass the Freedom to Vote Act. Pass the John Lewis Voting Rights Act. And while you're at it, pass the Disclose Act so Americans can know who is funding our elections.\n\nTonight, I'd like to honor someone who has dedicated his life to serve this country: Justice Stephen Breyer—an Army veteran, Constitutional scholar, and retiring Justice of the United States Supreme Court. Justice Breyer, thank you for your service. \n\nOne of the most serious constitutional
```

```
responsibilities a President has is nominating someone to  
serve on the United States Supreme Court. \n\nAnd I did that  
4 days ago, when I nominated Circuit Court of Appeals Judge  
Ketanji Brown Jackson. One of our nation's top legal minds,  
who will continue Justice Breyer's legacy of excellence.',  
metadata={'source': '.././../state_of_the_union.txt'})
```

Filtering on metadata

It can be helpful to narrow down the collection before working with it.

For example, collections can be filtered on metadata using the get method.

```
# filter collection for updated source  
example_db.get(where={"source": "some_other_source"})
```

```
{'ids': [], 'embeddings': None, 'metadatas': [], 'documents':  
[]}
```