

[Modules](#)[Retrieval](#)[Retrievers](#)[Vector store-backed retriever](#)

# Vector store-backed retriever

A vector store retriever is a retriever that uses a vector store to retrieve documents. It is a lightweight wrapper around the vector store class to make it conform to the retriever interface. It uses the search methods implemented by a vector store, like similarity search and MMR, to query the texts in the vector store.

Once you construct a vector store, it's very easy to construct a retriever. Let's walk through an example.

```
from langchain_community.document_loaders
import TextLoader

loader =
TextLoader("../state_of_the_union.txt")
```

```
from langchain.text_splitter import
CharacterTextSplitter
from langchain_community.vectorstores import
FAISS
from langchain_openai import OpenAIEmbeddings
```

```
documents = loader.load()
text_splitter =
CharacterTextSplitter(chunk_size=1000,
chunk_overlap=0)
texts =
text_splitter.split_documents(documents)
embeddings = OpenAIEmbeddings()
db = FAISS.from_documents(texts, embeddings)
```

```
retriever = db.as_retriever()
```

```
docs = retriever.get_relevant_documents("what
did he say about ketanji brown jackson")
```

## Maximum marginal relevance retrieval

By default, the vector store retriever uses similarity search. If the underlying vector store supports maximum marginal relevance search, you can specify that as the search type.

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

```
docs = retriever.get_relevant_documents("what  
did he say about ketanji brown jackson")
```

## Similarity score threshold retrieval

You can also set a retrieval method that sets a similarity score threshold and only returns documents with a score above that threshold.

```
retriever = db.as_retriever(  
    search_type="similarity_score_threshold",  
    search_kwargs={"score_threshold": 0.5}  
)
```

```
docs = retriever.get_relevant_documents("what  
did he say about ketanji brown jackson")
```

## Specifying top k

You can also specify search kwargs like `k` to use when doing retrieval.

```
retriever = db.as_retriever(search_kwargs=  
{"k": 1})
```

```
docs = retriever.get_relevant_documents("what  
did he say about ketanji brown jackson")  
len(docs)
```

```
1
```