Chains

Chains refer to sequences of calls - whether to an LLM, a tool, or a data preprocessing step. The primary supported way to do this is with LCEL.

LCEL is great for constructing your own chains, but it's also nice to have chains that you can use off-the-shelf. There are two types of off-the-shelf chains that LangChain supports:

- Chains that are built with LCEL. In this case, LangChain offers a higher-level constructor method. However, all that is being done under the hood is constructing a chain with LCEL.

- [Legacy] Chains constructed by subclassing from a legacy `Chain` class. These chains do not use LCEL under the hood but are rather standalone classes.

We are working creating methods that create LCEL versions of all chains. We are doing this for a few reasons.

1. Chains constructed in this way are nice because if you want to modify the internals of a chain you can simply

modify the LCEL.

2. These chains natively support streaming, async, and batch out of the box.

3. These chains automatically get observability at each step.

This page contains two lists. First, a list of all LCEL chain constructors. Second, a list of all legacy Chains.

LCEL Chains

Below is a table of all LCEL chain constructors. In addition, we report on:

Chain Constructor

The constructor function for this chain. These are all methods that return LCEL runnables. We also link to the API documentation.

Function Calling

Whether this requires OpenAI function calling.

Other Tools

What other tools (if any) are used in this chain.

When to Use

Our commentary on when to use this chain.

Chain Constructor	Function Calling	Other Tools
create_stuff_documents_chain		

Chain Constructor	Function Calling	Other Tools
create_openai_fn_runnable	✅	
create_structured_output_runnable	✅	

Chain Constructor	Function Calling	Other Tools
load_query_constructor_runnable		

Chain Constructor	Function Calling	Other Tools
create_sql_query_chain		SQL Database
create_history_aware_retriever		Retriever

Chain Constructor	Function Calling	Other Tools
create_retrieval_chain		Retriever

Chain Constructor	Function Calling	Other Tools

Legacy Chains

Below we report on the legacy chain types that exist. We will maintain support for these until we are able to create a LCEL alternative. We report on:

Chain

Name of the chain, or name of the constructor method. If constructor method, this will return a `Chain` subclass.

Function Calling

Whether this requires OpenAI Function Calling.

Other Tools

Other tools used in the chain.

When to Use

Our commentary on when to use.

Chain	Function Calling	Other To
APIChain		Requests Wrapper
OpenAPIEndpointChain		OpenAPI Spec
ConversationalRetrievalChain		Retriever

Chain	Function Calling	Other To
StuffDocumentsChain		
ReduceDocumentsChain		

Chain	Function Calling	Other To
MapReduceDocumentsChain		
RefineDocumentsChain		

Chain	Function Calling	Other To
MapRerankDocumentsChain		
ConstitutionalChain		

Chain	Function Calling	Other To
LLMChain		
ElasticsearchDatabaseChain		ElasticSea Instance
FlareChain		
ArangoGraphQAChain		Arango Graph

Chain	Function Calling	Other To
GraphCypherQAChain		A graph th works witl Cypher qu language
FalkorDBGraphQAChain		Falkor Database
HugeGraphQAChain		HugeGrap
KuzuQAChain		Kuzu Grap

Chain	Function Calling	Other To
NebulaGraphQAChain		Nebula Graph
NeptuneOpenCypherQAChain		Neptune Graph
GraphSparqlChain		Graph tha works wit SparQL

Chain	Function Calling	Other To
LLMMath		
LLMCheckerChain		
LLMSummarizationChecker		
create_citation_fuzzy_match_chain	✅	

Chain	Function Calling	Other To
create_extraction_chain	✅	
create_extraction_chain_pydantic	✅	
get_openapi_chain	✅	OpenAPI Spec
create_qa_with_structure_chain	✅	
create_qa_with_sources_chain	✅	
QAGenerationChain		

Chain	Function Calling	Other To
RetrievalQAWithSourcesChain		Retriever
load_qa_with_sources_chain		Retriever

Chain	Function Calling	Other To
RetrievalQA		Retriever
MultiPromptChain		
MultiRetrievalQAChain		Retriever
EmbeddingRouterChain		

Chain	Function Calling	Other To
LLMRouterChain		
load_summarize_chain		
LLMRequestsChain		