



# Router

This notebook demonstrates how to use the `RouterChain` paradigm to create a chain that dynamically selects the next chain to use for a given input.

Router chains are made up of two components:

- The RouterChain itself (responsible for selecting the next chain to call)
- `destination_chains`: chains that the router chain can route to

In this notebook we will focus on the different types of routing chains. We will show these routing chains used in a `MultiPromptChain` to create a question-answering chain that selects the prompt which is most relevant for a given question, and then answers the question using that prompt.

```
from langchain.chains.router import MultiPromptChain
from langchain.llms import OpenAI
from langchain.chains import ConversationChain
from langchain.chains.llm import LLMChain
from langchain.prompts import PromptTemplate
```



## API Reference:

- `MultiPromptChain` from `langchain.chains.router`
- `OpenAI` from `langchain.llms`
- `ConversationChain` from `langchain.chains`
- `LLMChain` from `langchain.chains.llm`
- `PromptTemplate` from `langchain.prompts`

```
physics_template = """You are a very smart physics professor. \
You are great at answering questions about physics in a concise and \
easy to understand manner. \
When you don't know the answer to a question you admit that you don't \
know.
```

```
Here is a question:

```

```
math_template = """You are a very good mathematician. You are great at
answering math questions. \
You are so good because you are able to break down hard problems into
their component parts, \
answer the component parts, and then put them together to answer the
broader question.
```

```
Here is a question:
{input}"""
```

```
prompt_infos = [
    {
        "name": "physics",
        "description": "Good for answering questions about physics",
        "prompt_template": physics_template,
    },
    {
        "name": "math",
        "description": "Good for answering math questions",
        "prompt_template": math_template,
    },
]
```

```
llm = OpenAI()
```

```
destination_chains = {}
for p_info in prompt_infos:
    name = p_info["name"]
    prompt_template = p_info["prompt_template"]
    prompt = PromptTemplate(template=prompt_template, input_variables=
["input"])
    chain = LLMChain(llm=llm, prompt=prompt)
    destination_chains[name] = chain
default_chain = ConversationChain(llm=llm, output_key="text")
```

## LLMRouterChain

---

This chain uses an LLM to determine how to route things.

```
from langchain.chains.router.llm_router import LLMRouterChain,
RouterOutputParser
from langchain.chains.router.multi_prompt_prompt import
MULTI_PROMPT_ROUTER_TEMPLATE
```

### API Reference:

- `LLMRouterChain` from `langchain.chains.router.llm_router`
- `RouterOutputParser` from `langchain.chains.router.llm_router`

```
destinations = [f"{p['name']}: {p['description']}" for p in
prompt_infos]
destinations_str = "\n".join(destinations)
router_template =
MULTI_PROMPT_ROUTER_TEMPLATE.format(destinations=destinations_str)
router_prompt = PromptTemplate(
    template=router_template,
    input_variables=["input"],
    output_parser=RouterOutputParser(),
)
router_chain = LLMRouterChain.from_llm(llm, router_prompt)
```

```
chain = MultiPromptChain(
    router_chain=router_chain,
    destination_chains=destination_chains,
    default_chain=default_chain,
    verbose=True,
)
```

```
print(chain.run("What is black body radiation?"))
```

```
> Entering new MultiPromptChain chain...
physics: {'input': 'What is black body radiation?'}
> Finished chain.
```

Black body radiation is the term used to describe the electromagnetic radiation emitted by a “black body”—an object that absorbs all radiation incident upon it. A black body is an idealized

physical body that absorbs all incident electromagnetic radiation, regardless of frequency or angle of incidence. It does not reflect, emit or transmit energy. This type of radiation is the result of the thermal motion of the body's atoms and molecules, and it is emitted at all wavelengths. The spectrum of radiation emitted is described by Planck's law and is known as the black body spectrum.

```
print(
    chain.run(
        "What is the first prime number greater than 40 such that one
        plus the prime number is divisible by 3"
    )
)
```

```
> Entering new MultiPromptChain chain...
math: {'input': 'What is the first prime number greater than 40
such that one plus the prime number is divisible by 3'}
> Finished chain.
?
```

The answer is 43. One plus 43 is 44 which is divisible by 3.

```
print(chain.run("What is the name of the type of cloud that rains"))
```

```
> Entering new MultiPromptChain chain...
None: {'input': 'What is the name of the type of cloud that
rains?'}
> Finished chain.
The type of cloud that rains is called a cumulonimbus cloud. It is
a tall and dense cloud that is often accompanied by thunder and
lightning.
```

## EmbeddingRouterChain

The EmbeddingRouterChain uses embeddings and similarity to route between destination chains.

```
from langchain.chains.router.embedding_router import
EmbeddingRouterChain
from langchain.embeddings import CohereEmbeddings
from langchain.vectorstores import Chroma
```

### API Reference:

- `EmbeddingRouterChain` from `langchain.chains.router.embedding_router`
- `CohereEmbeddings` from `langchain.embeddings`
- `Chroma` from `langchain.vectorstores`

```
names_and_descriptions = [
    ("physics", ["for questions about physics"]),
    ("math", ["for questions about math"]),
]
```

```
router_chain = EmbeddingRouterChain.from_names_and_descriptions(
    names_and_descriptions, Chroma, CohereEmbeddings(), routing_keys=
["input"]
)
```

Using embedded DuckDB without persistence: data will be transient

```
chain = MultiPromptChain(
    router_chain=router_chain,
    destination_chains=destination_chains,
    default_chain=default_chain,
    verbose=True,
)
```

```
print(chain.run("What is black body radiation?"))
```

```
> Entering new MultiPromptChain chain...
physics: {'input': 'What is black body radiation?'}
> Finished chain.
```

Black body radiation is the emission of energy from an idealized physical body (known as a black body) that is in thermal equilibrium with its environment. It is emitted in a characteristic pattern of frequencies known as a black-body spectrum, which depends only on the temperature of the body. The study of black body radiation is an important part of astrophysics and atmospheric physics, as the thermal radiation emitted by stars and planets can often be approximated as black body radiation.

```
print(
    chain.run(
        "What is the first prime number greater than 40 such that one
        plus the prime number is divisible by 3"
    )
)
```

```
> Entering new MultiPromptChain chain...
math: {'input': 'What is the first prime number greater than 40
such that one plus the prime number is divisible by 3'}
> Finished chain.
?
```

Answer: The first prime number greater than 40 such that one plus the prime number is divisible by 3 is 43.