Interface

In an effort to make it as easy as possible to create custom chains, we've implemented a "Runnable" protocol that most components implement. This is a standard interface with a few different methods, which makes it easy to define custom chains as well as making it possible to invoke them in a standard way. The standard interface exposed includes:

- `stream`: stream back chunks of the response
- `invoke`: call the chain on an input
- `batch`: call the chain on a list of inputs

These also have corresponding async methods:

- `astream`: stream back chunks of the response async
- `ainvoke`: call the chain on an input async
- `abatch`: call the chain on a list of inputs async

The type of the input varies by component:

Component	Input Type
Prompt	Dictionary
Retriever	Single string
Model	Single string, list of chat messages or a PromptValue

The output type also varies by component:

Component	Output Type
LLM	String
ChatModel	ChatMessage
Prompt	PromptValue

Component	Output Type
Retriever	List of documents

Let's take a look at these methods! To do so, we'll create a super simple PromptTemplate + ChatModel chain.

```
from langchain.prompts import ChatPromptTemplate
from langchain.chat_models import ChatOpenAI
```

API Reference:

- ChatPromptTemplate from `langchain.prompts`
- ChatOpenAI from `langchain.chat_models`

```
model = ChatOpenAI()
```

```
prompt = ChatPromptTemplate.from_template("tell me a joke about {topic}")
```

```
chain = prompt | model
```

Stream

```
for s in chain.stream({"topic": "bears"}):
    print(s.content, end="", flush=True)
```

```
    Sure, here's a bear-themed joke for you:

    Why don't bears wear shoes?

    Because they have bear feet!
```

Invoke

```
chain.invoke({"topic": "bears"})
```

```
    AIMessage(content="Why don't bears wear shoes?\n\nBecause they
already have bear feet!", additional_kwargs={}, example=False)
```

Batch

```
chain.batch([{"topic": "bears"}, {"topic": "cats"}])
```

```
    [AIMessage(content="Why don't bears ever wear shoes?\n\nBecause
they have bear feet!", additional_kwargs={}, example=False),
     AIMessage(content="Why don't cats play poker in the wild?\n\nToo
many cheetahs!", additional_kwargs={}, example=False)]
```

Async Stream

```
async for s in chain.astream({"topic": "bears"}):
    print(s.content, end="", flush=True)
```

```
    Why don't bears wear shoes?

    Because they have bear feet!
```

Async Invoke

```
await chain.ainvoke({"topic": "bears"})
```

```
    AIMessage(content="Sure, here you go:\n\nWhy don't bears wear
shoes?\n\nBecause they have bear feet!", additional_kwargs={},
```

```
example=False)
```

Async Batch

```
await chain.abatch([{"topic": "bears"}])
```

```
    [AIMessage(content="Why don't bears wear shoes?\n\nBecause they
have bear feet!", additional_kwargs={}, example=False)]
```

Parallelism

Let's take a look at how LangChain Expression Language support parralel requests as much as possible. For example, when using a RunnableMapping (often written as a dictionary) it executes each element in parralel.

```
from langchain.schema.runnable import RunnableMap
chain1 = ChatPromptTemplate.from_template("tell me a joke about
{topic}") | model
chain2 = ChatPromptTemplate.from_template("write a short (2 line) poem
about {topic}") | model
combined = RunnableMap({
    "joke": chain1,
    "poem": chain2,
})
```

API Reference:

- RunnableMap from `langchain.schema.runnable`

```
chain1.invoke({"topic": "bears"})
```

```
    CPU times: user 31.7 ms, sys: 8.59 ms, total: 40.3 ms
    Wall time: 1.05 s
```

```
    AIMessage(content="Why don't bears like fast food?\n\nBecause they
can't catch it!", additional_kwargs={}, example=False)
```

```
chain2.invoke({"topic": "bears"})
```

```
    CPU times: user 42.9 ms, sys: 10.2 ms, total: 53 ms
    Wall time: 1.93 s
```

```
    AIMessage(content="In forest's embrace, bears roam free,\nSilent
strength, nature's majesty.", additional_kwargs={}, example=False)
```

```
combined.invoke({"topic": "bears"})
```

```
    CPU times: user 96.3 ms, sys: 20.4 ms, total: 117 ms
    Wall time: 1.1 s
```

```
    {'joke': AIMessage(content="Why don't bears wear socks?\n\nBecause
they have bear feet!", additional_kwargs={}, example=False),
     'poem': AIMessage(content="In forest's embrace,\nMajestic bears
leave their trace.", additional_kwargs={}, example=False)}
```