Apify

This notebook shows how to use the Apify integration for LangChain.

Apify is a cloud platform for web scraping and data extraction, which provides an ecosystem of more than a thousand ready-made apps called *Actors* for various web scraping, crawling, and data extraction use cases. For example, you can use it to extract Google Search results, Instagram and Facebook profiles, products from Amazon or Shopify, Google Maps reviews, etc. etc.

In this example, we'll use the Website Content Crawler Actor, which can deeply crawl websites such as documentation, knowledge bases, help centers, or blogs, and extract text content from the web pages. Then we feed the documents into a vector index and answer questions from it.

First, import `ApifyWrapper` into your source code:

```
from langchain.document_loaders.base import Document
from langchain.indexes import VectorstoreIndexCreator
from langchain.utilities import ApifyWrapper
```

Initialize it using your Apify API token and for the purpose of this example, also with your OpenAI API key:

```
import os
os.environ["OPENAI_API_KEY"] = "Your OpenAI API key"
os.environ["APIFY_API_TOKEN"] = "Your Apify API token"

apify = ApifyWrapper()
```

Then run the Actor, wait for it to finish, and fetch its results from the Apify dataset into a LangChain document loader.

Note that if you already have some results in an Apify dataset, you can load them directly using `ApifyDatasetLoader`, as shown in this notebook. In that notebook, you'll also find the explanation of the `dataset_mapping_function`, which is used to map fields from the Apify

Skip to main content

```python
loader = apify.call_actor(
    actor_id="apify/website-content-crawler",
    run_input={"startUrls": [{"url": "https://python.langchain.com/en/latest/"}]},
    dataset_mapping_function=lambda item: Document(
        page_content=item["text"] or "", metadata={"source": item["url"]}
    ),
)
```

Initialize the vector index from the crawled documents:

```python
index = VectorstoreIndexCreator().from_loaders([loader])
```

And finally, query the vector index:

```python
query = "What is LangChain?"
result = index.query_with_sources(query)
```

```python
print(result["answer"])
print(result["sources"])
```

```
 LangChain is a standard interface through which you can interact with a variety
of large language models (LLMs). It provides modules that can be used to build
language model applications, and it also provides chains and agents with memory
capabilities.

https://python.langchain.com/en/latest/modules/models/llms.html,
https://python.langchain.com/en/latest/getting_started/getting_started.html
```