

# Cache management

In the world of data analysis and processing, efficiency and speed are paramount. This is where caching comes into play. Caching is a powerful technique used to temporarily store copies of data, making future requests for that data faster and reducing the need to repeat expensive operations. It's particularly beneficial when dealing with large datasets, frequent queries, or complex computations that are resource-intensive to compute repeatedly.

Throughout this tutorial, we'll explore the key functionalities Giza offers for cache management.

How it works

The default cache directory is `~/.cache/giza_datasets`

when you are creating your `DatasetLoader` object, you have the option to modify the path of your cache:

```
...
```

```
Copy from giza_datasets import DatasetLoader loader=DatasetLoader(cache_dir=".")
```

```
...
```

or disable it:

```
...
```

```
Copy loader=DatasetLoader(use_cache=False)
```

```
...
```

and it would be done! With this simple configuration Giza takes care of downloading, saving and uploading the necessary data efficiently.

Finally, if you want to clear the cache, you can run the following command:

```
...
```

```
Copy loader.clear_cache()# 1 datasets have been cleared from the cache directory.
```

```
...
```

[Previous Eager mode](#) [Next Aggregated datasets](#)

Last updated 1 day ago