Python Programming

THE PERFECT WAY TO CODE YOUR FUTURE

Cache Django

ENTRI elevate

What is caching?

- Caching in Python Django refers to the practice of temporarily storing and reusing computed or fetched data to improve the performance and response time of web applications. Caching helps reduce the workload on the server, as it allows frequently accessed data to be served quickly without the need to regenerate or re-fetch it from the source.

Common use cases and concepts related to caching

- **Database Query Results:** One of the most common use cases for caching in Django is to cache the results of database queries. Instead of querying the database every time a particular request is made, you can cache the query results and serve them from the cache if the same query is made within a specified period.

- **Rendered HTML Pages:** You can cache the HTML output of rendered templates to avoid the need to regenerate the HTML for each request. This is particularly useful for pages that don't change frequently.
- **API Responses:** Caching can be applied to the responses of API endpoints. This reduces the load on your server when serving frequently requested API data.
- **Session and Authentication Data:** Caching can be used to store session data and authentication tokens, reducing the need to hit the database or perform complex computations for every request.

- **Full-page Caching:** Django provides a middleware called the "cache middleware" that can cache entire rendered HTML pages, serving them to subsequent users without re-rendering the page.
- **Low-Level Cache API:** Django's caching framework includes a low-level cache API that allows you to cache arbitrary data such as function results, serialized objects, or any Python data structure.

To use caching in Django, you need to configure it in your project's settings by specifying the cache backend you want to use (e.g., memory-based cache, file-based cache, Redis, Memcached). Here's an example of how you might configure caching in your Django settings:

python Copy code

```python
# settings.py

CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': '127.0.0.1:11211',
    }
}
```


Thank
You!