**Memcached Caching Demo**

# **Overview**

Shape

This document provides instructions on setting up Memcached caching in a Laravel application. Additionally, a simple web interface is created to demonstrate how to set and get cached data through a user-friendly form with input fields.

The demo will use Memcached to store and retrieve data for a short period (10 minutes), showcasing Laravel's caching system and the Memcached cache driver.

# **Usage**

Open the browser and visit http://127.0.0.1:8000/cache.

* You will see two forms:
* Set Cache: To store a cache value by providing a key and value.
* Get Cache: To retrieve a cached value by providing a key.
* Set Cache:
* Input a cache key and value in the provided fields, then click "Set Cache."
* A success message will confirm that the data has been cached.
* Get Cache:
* Input the cache key in the "Enter Cache Key" field, then click "Get Cache."
* If the cache exists, the cached value will be displayed.
* If the cache has expired or doesn't exist, the message "Cache not found" will be shown.

# **Prerequisites**

Shape

### 1. **Laravel Installation**

Ensure you have a Laravel application set up. You can create a new Laravel application using the following command:

composer create-project --prefer-dist laravel/laravel laravel-memcached-demo

### 2. **Memcached Server Installation**

Memcached must be installed and running on your local machine or server. For Ubuntu, you can install it as follows:

sudo apt-get install memcached

sudo systemctl start memcached

sudo systemctl enable memcached

For other systems, you can refer to the [Memcached official installation guide](https://memcached.org/).

### **3. Install PHP Memcached Extension**

Make sure the Memcached extension is installed for PHP. On Ubuntu:

sudo apt-get install php-memcached

sudo service apache2 restart

# **Benefits**

* Performance Boost: Memcached can significantly improve the speed of your Laravel application by caching expensive queries or frequent data.
* Ease of Use: The demo uses a simple web interface for interacting with the cache, making it easy to test and understand caching concepts.