# Spring Cache

# What Is Spring Cache?

‣ A caching abstraction provided by Spring Framework.

‣ Helps reduce expensive operations (like DB calls or computations).

‣ Transparent: Just annotate methods; Spring handles the caching.

‣ Supports multiple providers (e.g., ConcurrentMap, EhCache, Redis, Caffeine).

# Why Use Caching in Spring?

▸ Improved performance – Avoid repeated costly operations.

▸ Declarative and flexible – Annotations make caching easy to control.

▸ Out-of-the-box support – Comes integrated with Spring Boot Starter.

▸ Extensible – Switch providers without code changes.

# Key Annotations

▸ @EnableCaching – Activates Spring's caching support.

▸ @Cacheable – Marks a method to cache the result.

▸ @CacheEvict – Removes entries from the cache.

▸ @CachePut – Updates the cache with a fresh value.

# How Spring Cache Works

‣ 1. Spring checks the cache before method execution.

‣ 2. If a cached value is found, it's returned directly.

‣ 3. Otherwise, the method runs, and the result is stored in the cache.

# Demo 1 - Spring Cache

▸ Using Spring Cache

▸ Using annotations like @EnableCaching, @Cacheable and @CacheEvict

# Exercise 1.1 - Spring Cache

▸ Reuse the solution from Managing Beans And Dependency Injection

▸ In the sum method of the Calculator class, add code like this example before returning from the method to make the method less efficient:

▸ Then run the project and call a Controller method and it should take at least 5 seconds to reply

```java
try {
    Thread.sleep( millis: 5000);
} catch (InterruptedException e) {
    throw new RuntimeException(e);
}
```

# Exercise 1.2 - Spring Cache

‣ Now speed up the functionality by enabling caching to the sum method

‣ Add the @EnableCaching annotation to the Calculator class

‣ Add the @Cacheable("myCache") annotation to the sum method

‣ Now try to run the application and you should notice that the first time you calculate two numbers it takes 5 seconds, but the second time you calculate the same numbers it will now go very fast since the return value of these two numbers is now automatically cached