
Pivotal®

Caching with Spring

Spring's Cache Abstraction

John Blum
Spring Data Team

Pivotal

Why Caching?

Caching is beneficial when an **expensive** operation (i.e. CPU / IO bound) produces the **same output** given **identical input** (the result of the operation can be reused).

Examples:

- Eligibility Determination
  - Involving complex rules or calculations
- Routing
  - Computing the quickest or most efficient route between point A and B (Shortest Path based on Weighted/Directed Graph)
- Network I/O
  - Modern Enterprise Applications are a mashup of many different independent and desperate Microservices all interoperating to achieve their objective.

Pivotal

How does **Spring** help?

“Simple things should be simple; complex things should be possible”

– Alan Kay

Pivotal

Spring Cache Abstraction

Declarative, Annotation-based Caching

- `@Cacheable` – caches result of a method
- `@CacheEvict` – triggers a cache eviction
- `@CachePut` – updates cache without interfering with method execution
- `@Caching` – groups multiple cache operations per method
- `@CacheConfig` – class-level cache settings

Pivotal

JSR-107 - JCache

Spring recognizes JSR-107 Annotations

Spring	JCache
`@Cacheable`	`@CacheResult`
`@CachePut`	`@CachePut`
`@CacheEvict`	`@CacheRemove`
`@CacheEvict(allEntries=true)`	`@CacheRemoveAll`
`@CacheConfig`	`@CacheDefaults`

Pivotal

Spring Cache Abstraction

Custom Key Generation

```
@Cacheable(cacheNames = "Books", key = "#isbn")
public Book findBook(ISBN isbn, boolean checkWarehouse, boolean used) {
  …
}

@Cacheable(cacheNames = "Books", key = "T(someType).hash(#isbn)")
public Book findBook(ISBN isbn, boolean checkWarehouse, boolean used) {
  …
}

@Cacheable(cacheNames = "Books", keyGenerator = "myKeyGeneratorBeanName")
public Book findBook(ISBN isbn, boolean checkWarehouse, boolean used) {
  …
}
```

Pivotal

Spring Cache Abstraction

Synchronized Caching

```
@Cacheable(cacheNames = "Books", sync = "true")
public Set<Book> findBooks(Author author, DateRange dateRange) {
  …
}
```

Pivotal

Spring Cache Abstraction

Conditional Caching

```
@Cacheable(cacheNames = "Books", condition = "#author.reputation > 65"
    unless="#result?.print")
Public Book findBook(Author author, String title) {
  …
}
```

Demo

Spring Data GemFire / Geode

Spring Framework

Using **Pivotal GemFire** or **Apache Geode**
With *Spring's* Cache Abstraction and *Spring Data* GemFire/Geode
To serve as a **JCache** (JSR-107) caching provider

Spring Data GemFire

Spring Session

Using **Pivotal GemFire** with Spring Session (Data GemFire)
To simplify (HTTP) **Session State Management**

Spring Data GemFire

Spring Boot

Using **Pivotal GemFire** with **Spring Boot**
To **get up and running** as **quickly** as possible

Pivotal

Questions

Answers

Pivotal

Thank you

Pivotal


Pivotal®