

06-Web Caching By Server

, /
.
Read ,
.

- -
 - CACHE
 -
- ()
 -
 - 1.
 - 2.
- - Caffeine
 -
 -
 - DB
-

Write-through with no-write-allocation	Write-back with write-allocation
blocked URL	blocked URL

: [https://en.wikipedia.org/wiki/Cache_\(computing\)](https://en.wikipedia.org/wiki/Cache_(computing))

, .
?
.

CACHE

- ()
- Proxy, (Proxy,)
- () ()

/

.

-
- Redis DB ,

()

```

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-cache</artifactId>
</dependency>

```

1.

Application @EnableCaching

```

@SpringBootApplication
@EnableCaching
public class Application {
    .....
}

```

2.

```

package com.example.demo.data2;

import org.springframework.cache.annotation.Cacheable;
import org.springframework.stereotype.Component;

@Component
public class SimpleUserRepository {
    @Cacheable("books")
    public String getByIsbn(String isbn) throws Exception {
        long time = 3000L;
        Thread.sleep(time);
        return "Some String-" + isbn;
    }
}

```

Cacheable , .()

getByIsbn IO (3)

- getByIsbn('1') - 3
- getByIsbn('1') -
- getByIsbn('1') -
- getByIsbn('2') - 3
- getByIsbn('2') -
- getByIsbn('1') -

- ?
- ?
- ?
- ?

- .
- .
- , . ()

Caffeine

```
# application.property
spring.cache.cache-names: instruments, directory
spring.cache.caffeine.spec: maxSize=1000, expireAfterAccess=10s // (s) (m) (d)
logging.level.com.memorynotfound=info

# pom.xml
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-cache</artifactId>
</dependency>

<!-- caching provider -->
<dependency>
  <groupId>com.github.ben-manes.caffeine</groupId>
  <artifactId>caffeine</artifactId>
</dependency>
```

JPA Repository .

instruments, directory

10 1000 .

, 10 .

```
public class InputOpt {
    .....
    private int ecode;
    public int getEcode() {
        return hashCode();
    }
    @Override
    public int hashCode() {
        HashCodeBuilder builder = new HashCodeBuilder();
        builder.append(category1);
        builder.append(router);
        builder.append(action);
        builder.append(sort);
        builder.append(sortdir);
        builder.append(size);
        builder.append(userprofile);
        return builder.toHashCode();
    }
}
```

HashCodeBuilder . ecode

```
//
public interface GoodsDataService {
    Iterable<GoodsDataRC> findAll( BooleanBuilder searchOpt , Pageable pageRequest , InputOpt inputOpt
);
}

-----
//
@Service
@CacheConfig(cacheNames = {"directory", "instruments"})
public class GoodsDataServiceImpl implements GoodsDataService{

    static final Logger logger = LoggerFactory.getLogger(GoodsDataService.class);
    @Autowired
    private GoodsDataRCRepo      goodsDataRCRepo; //      JPA Repository

    @Cacheable( value="instruments", key="#inputOpt.ocode" )
    public Iterable<GoodsDataRC> findAll( BooleanBuilder searchOpt , Pageable pageRequest , InputOpt inputOpt )
    {
        logger.info("hash == %d",inputOpt.hashCode() );
        return goodsDataRCRepo.findAll(searchOpt,pageRequest);
    }
}
```

DB findAll . instruments

inputOpt.ocode , .

DB

```
@Autowired
GoodsDataService service;                //for cache db
.....
Iterable<GoodsDataRC> dbResult = service.findAll(searchOpt,pageRequest,inputOpt);
```

,

JPA DB .

Api inputOpt .

? JPA sql

sql .

:EndPoint

<http://javasampleapproach.com/spring-framework/cache-data-spring-cache-using-spring-boot>



DB (+-5) .

- A 100 (0)
- B 100 (1)
- C 100 (2)

.

.

? DB

,

, .

, .