

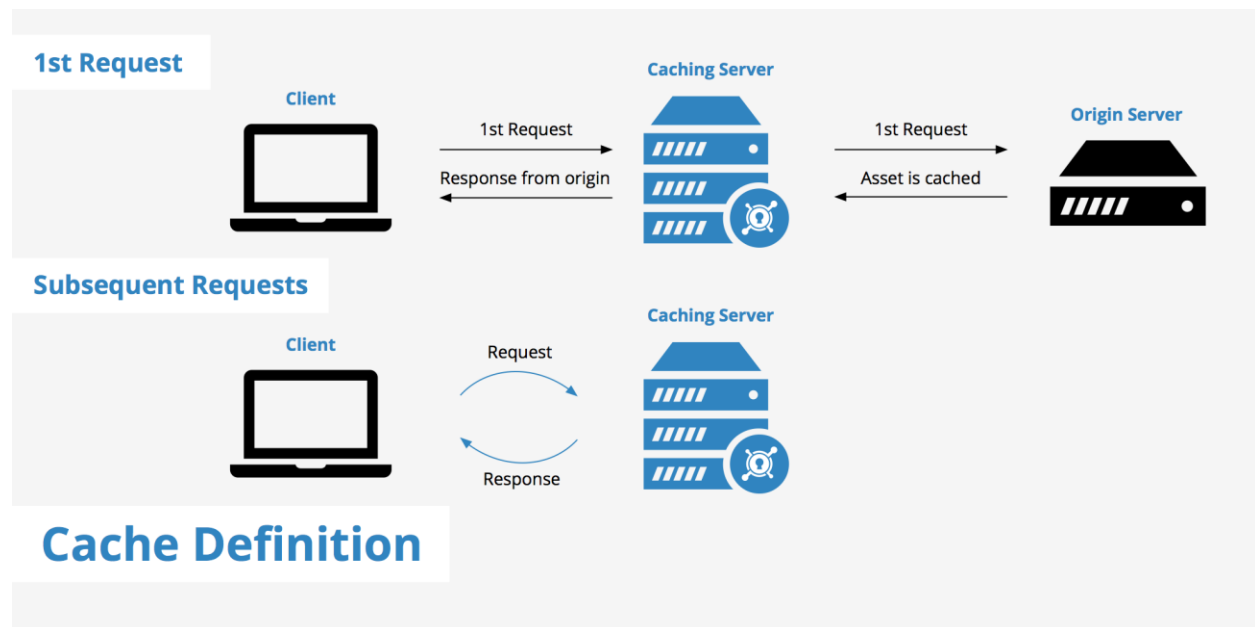
Contents

- Redis
- Cache , Cookie and the difference
- Server Cache vs Browser Cache
- Local Redis Server
- Online Centralized Redis Server



REDIS : Redis is an open source (BSD licensed), in-memory data structure store used as a database, cache, message broker, and streaming engine.

CACHE : A faster and smaller segment of memory whose access time is as close as registers are known as Cache memory. In a hierarchy of memory, cache memory has access time lesser than primary memory. Generally, cache memory is very small and hence is used as a buffer.



Need of cache memory

Data in primary memory can be accessed faster than secondary memory but still, access times of primary memory are generally in a few microseconds, whereas CPU is capable of performing operations in nanoseconds. Due to the time lag between accessing data and acting on data performance of the system decreases as the CPU is not utilized properly, it may remain idle for some time. In order to minimize this time gap a new segment of memory is Introduced known as Cache Memory.

How does cache work?

In order to understand the working of cache we must understand few points:

Cache memory is faster, they can be accessed very fast

Cache memory is smaller, a large amount of data cannot be stored

Whenever the CPU needs any data it searches for corresponding data in the cache (fast process) if data is found, it processes the data according to instructions, however, if data is not found in the cache CPU searches for that data in primary memory (slower process) and loads it into the cache. This ensures frequently accessed data is always found in the cache and hence minimizes the time required to access the data.

References :

<https://www.geeksforgeeks.org/cache-memory/>

<https://www.geeksforgeeks.org/cache-memory-in-computer-organization/>

What are Cookies?

Cookies are small files that contain information useful to a web site — such as password, preferences, browser, IP Address, date and time of visit, etc. Every time the user loads the website, the browser sends the cookie back to the server to notify the website of the user's previous activity.

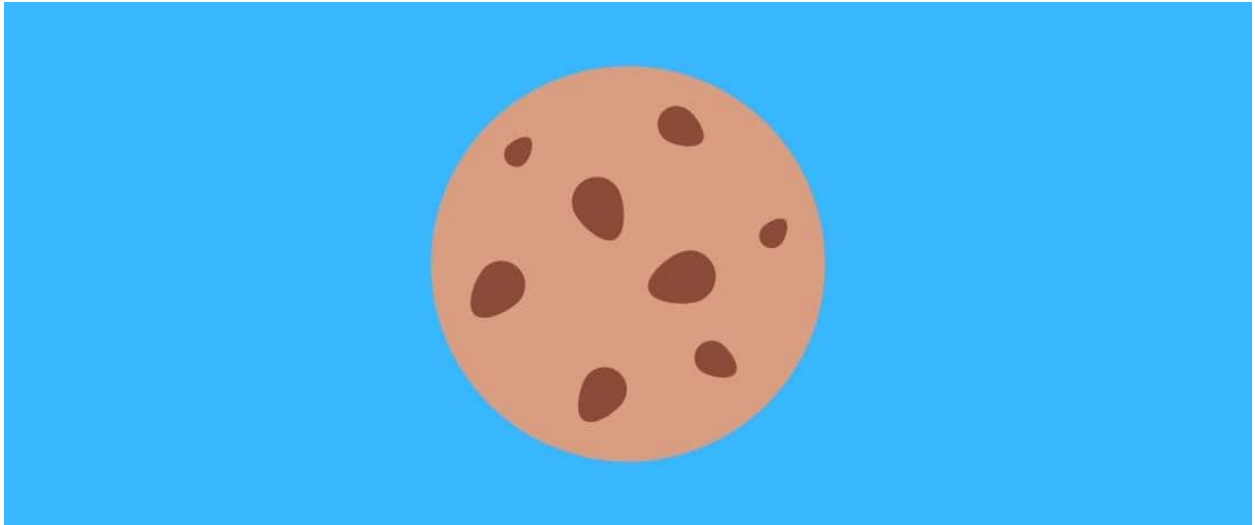
Cookies have a certain life span defined by their creators and it expires after the fixed time span.

Cookies often track information like how frequently the user visits, what are the times of visits, what banners have been clicked on, what button clicked, user preferences, items in shopping cart, etc. This allows the site to present you with information customized to fit your needs.

Cookies are usually used to store information needed for shorter periods. Cookies were first introduced by Netscape. In those earlier stages cookies did not receive good acceptance, since rumors said it might hack your personal data. Later people realized that cookies are actually harmless, and now they are highly accepted.

What's The Difference Between Cache And Cookies ?

- Cookies are used to store information to track different characteristics related to user, while cache is used to make the loading of web pages faster.
- Cookies store information such as user preferences, while cache will keep resource files such as audio, video or flash files.
- Typically, cookies expire after some time, but cache is kept in the client's machine until they are removed manually by the user.



Example of cache and cookies

Let's take a small example to clear the difference between cache and cookies. Suppose you visit a shopping website to check or buy a pair of shoes. You check some of the shoes of different brands or the same. After that either you buy or you leave the website without buying.

And now you go to any other website, whether that be a news website or any other which serves ads. You will notice that all the products you checked for are now following you in terms of ads. At times it could be a different product from a different website or same product from the same website. This is done by the help of cookies.

Now let's check out the role of cookies in it. What cookies do is to store information about your browsing history. In this case it stores which website you browsed and what kind of product you looked for. That information now helps the advertisers to show you the same product or related product in their ads.

The role of cache in this example is to store a small part of the image of the products you browsed and other web sources. Now, when you re-visit that website or product the page or the image will load faster. Because a small part of images and other web resources are already stored on your computer. So it helps in boosting the loading time.

References :

<https://www.geeksforgeeks.org/difference-between-cache-and-cookies/>

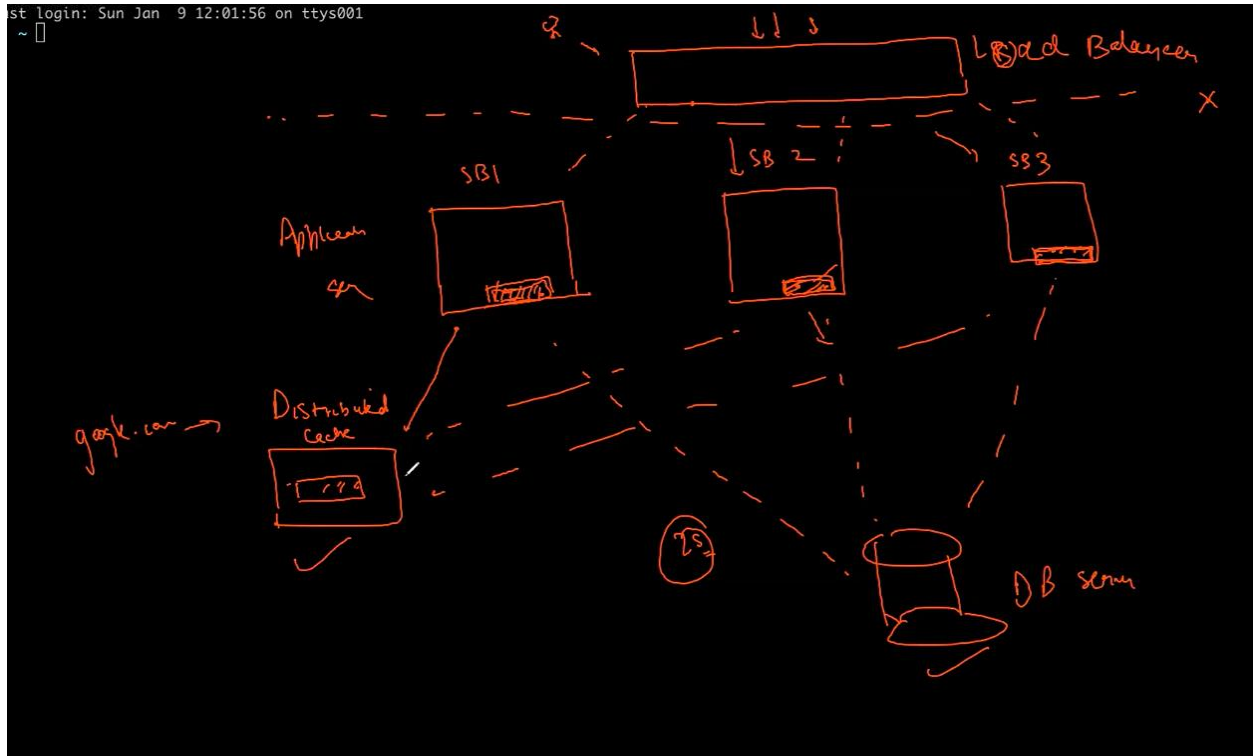
SERVER CACHE VS CLIENT CACHE :

A server cache is all about storing the mostly used data in the server. The server holds the pre-assembled version of the various web pages of a website.

Coming to the browser cache, the required data is stored in the user's hard drive. This is helpful as the user's computer does not have to download certain heavy elements of a web page again and again every time the user visits the webpage. However, a user has the right to clear the browser cache, and it is required when you want the server to serve the updated version of the webpage.

A server cache can hold content, code, queries, and that too on multiple servers to serve end users faster. On the other hand, the browser cache stores HTML pages, CSS files, JavaScript scripts, and most important images and other types of media objects. The media objects take the most time to load as they are large in size and therefore, if they are stored in the user's computer, it can be served instantly rather than bringing them from the server.

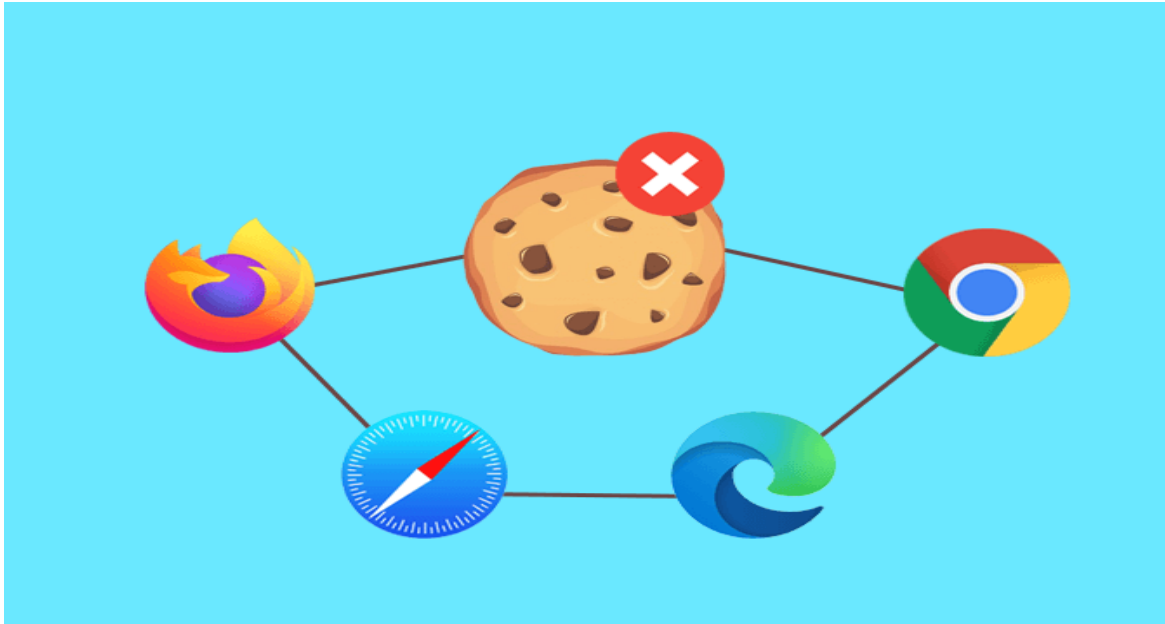
Client Cache can be found in the local storage tab .



Types of Server Cache :

-In Memory Cache

-Distributed Cache



Challenge 1 : How is a Distributed cache faster than Database ?

Ways of connecting to Redis :

1. Locally in the machine where you have your Spring Boot server
2. Cloud solution provided by Redis Labs

How to start the local redis server and the redis-cli?

Start the server by typing the following command in terminal :

```
>redis-server (mac)
```

```
><path of redis server>/redis-server.sh (windows)
```

Let us open redis cli shell with the command :

Session 13



```
>redis-cli
```

Redis stores data in key , value pairs . The only constraint is that key is always going to be a string.Its like a `HashMap<String,Object>` .

In redis-cli , lets see the keys present :

```
>keys *
```

Challenge 1: If we create a key

```
set key1 val1
```

We can get the value of key1 by :

```
get key1
```

Now if I stop the redis server and then start it again .

Is my data lost and what will I get when I do the below ?

```
get key1
```

Challenge 2: From the above experiment , Is Redis acting as Cache or Persistent Store ?

HINT : Redis is single threaded on the frontend . It can connect to only one client at one time .

Challenge 3: What is RDB and what is its location ?

HINT : search for your redis config file .

```
vi ../../usr/local/etc/redis.conf
```


Challenge 4: What kind of values can we store in a key in redis ?

-String , List and Hashes

Challenge 5: Set a key with an expiry of 45 seconds .

Note : Once it expires , it will show ttl (time to live as -2)

```
127.0.0.1:6379> set k3 v3 ex 45
OK
127.0.0.1:6379> get k3
"v3"
127.0.0.1:6379> get k3
"v3"
127.0.0.1:6379> ttl k3
(integer) 19
127.0.0.1:6379> ttl k3
(integer) 15
127.0.0.1:6379> ttl k3
(integer) 13
127.0.0.1:6379> ttl k3
(integer) 3
127.0.0.1:6379> ttl k3
(integer) -2
127.0.0.1:6379> □
```

Challenge 6: What is ttl for a key which has no expiry set ?

HINT: -1

Challenge 7: What is px in the following command :

`set k5 v5 px 5000`

HINT : px means milliseconds . This k5 will expire after 5000 milliseconds ie 5s .

Session 13

Challenge 8: We are setting a key myname with value aadhar .

`set myname aadhar`

How to expire this key when it was not set with an expiry ?

HINT : `expire myname 40`

Challenge 9: What is ttl for a key which never existed ?

HINT : `-2`

Challenge 10: How can we add a Redis List to key as value ?

HINT : `lpush players "Sachin" "Virat"`

Challenge 11: How to fetch the value of the key players list ?

HINT : `lrange players 0 2`

1)"Virat"

2)"Sachin"

Note : 2(end index) is inclusive above .

Challenge 12: Now how to see the whole list till the last index ?

HINT : `lrange players 0 -1`

Note : `lrange players 0 -2`

-> gives till second last index

Challenge 13: How to remove everything from the list ?

Session 13



HINT : `lpop players 2`

(---> here 2 is the total number of elements in list)

Challenge 14: What will happen if we try to push a non string value inside the list as below :

`lpush players "Aadhar" "Satish" 2 -1 true`

Will the above command give error ?

Challenge 15: How to store a Redis Hash as value for our key ?

HINT: A Hash in Redis is like a Map . It is a collection of field value pairs .

`hset person_key id 1 name Satish height 1.8`

Challenge 16: How to get the hash value of person_key ?

HINT : `hgetall person_key`

Challenge 17: What if I only want to get a single field of the hash value stored at person_key ?

HINT : `hget person_key name`

Challenge 18: What if you want to get more than one field of the hash value stored at person_key ?

HINT : `hmget person_key id name`

Challenge 19: What is the use of incr and decr when the values are stored as string ?

What will be the result in case 1 , 2 and 3 ?

Session 13



(1)

```
set k1 10
```

```
incr k1
```

(2)

```
set k1 v1
```

```
incr k1
```

(3)

```
set k1 20
```

```
decr k1
```

Challenge 20: What are the disadvantages of the local Redis server ? How to solve the issue ?

HINT : Distributed Central Cache

Challenge 21: Which is better type of cache : In memory or Central ?

HINT : There is no better or worse . It is situational depending on how requests are routed through Load Balancer .

Challenge 22: How to connect to the Redis Labs Central Cache ?

HINT : Create a redis labs free tier account . You will get 30 MB free . Create a new database in the central server . Now to connect to this central redis server database , you need to give your host and port . You will get them from your public endpoint .

```
redis-cli -h <host> -p <port> -a <default user password>
```

Session 13

Now you will be connected .

Now you can access keys and add keys as per your desire .

You can view your created keys in the metric tab of the redis dashboard website .

Note : To learn about redis commands : `redis-cli --help`

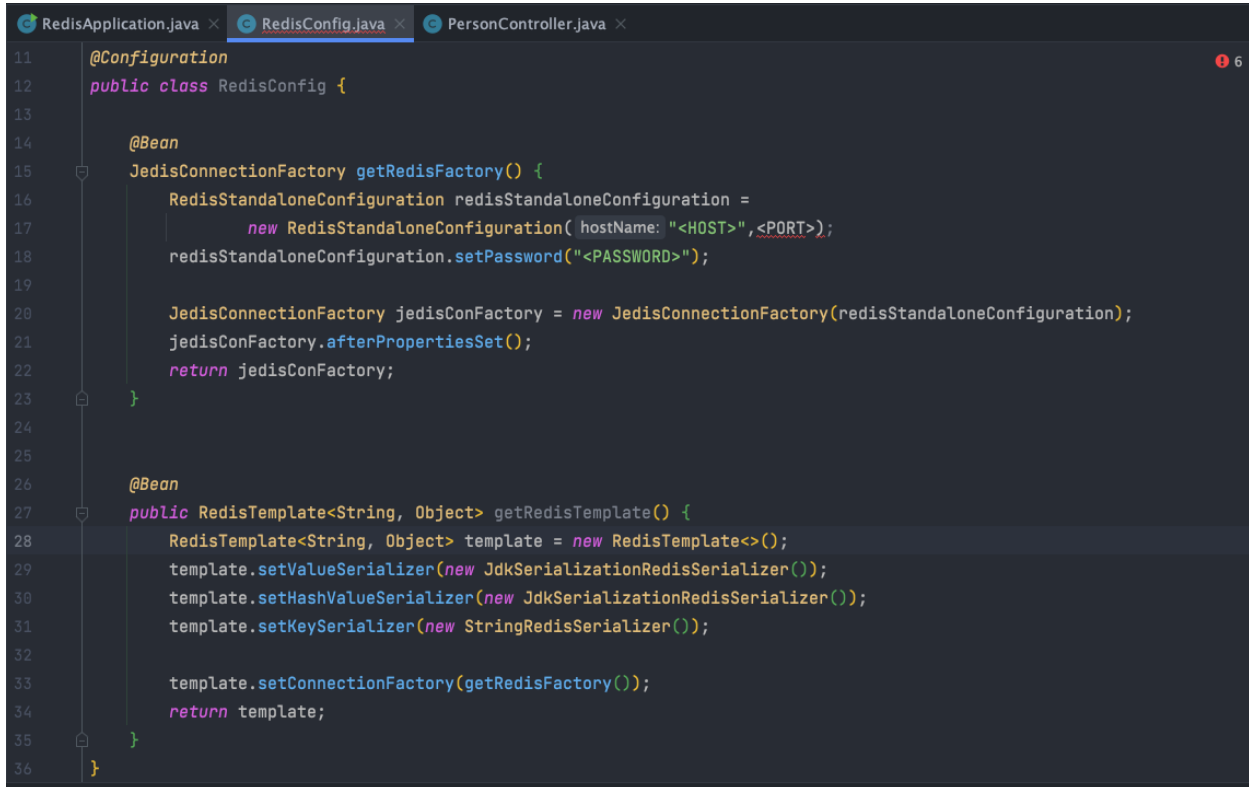
Challenge 23: How to connect the Redis Central cache server to your Spring Boot server . No need of persisting in the database .



HINT :

1. Create a new maven spring boot project .
2. Add the following dependencies :
 - lombok
 - spring web
 - spring data redis(Access + Driver)
 - exclude lettuce from spring data redis
 - add jedis

3. Now create the Redis Config class for configuration of Redis Factory and Redis Template .



```
11 @Configuration
12 public class RedisConfig {
13
14     @Bean
15     JedisConnectionFactory getRedisFactory() {
16         RedisStandaloneConfiguration redisStandaloneConfiguration =
17             new RedisStandaloneConfiguration( hostName: "<HOST>", <PORT>);
18         redisStandaloneConfiguration.setPassword("<PASSWORD>");
19
20         JedisConnectionFactory jedisConFactory = new JedisConnectionFactory(redisStandaloneConfiguration);
21         jedisConFactory.afterPropertiesSet();
22         return jedisConFactory;
23     }
24
25
26     @Bean
27     public RedisTemplate<String, Object> getRedisTemplate() {
28         RedisTemplate<String, Object> template = new RedisTemplate<>();
29         template.setValueSerializer(new JdkSerializationRedisSerializer());
30         template.setHashValueSerializer(new JdkSerializationRedisSerializer());
31         template.setKeySerializer(new StringRedisSerializer());
32
33         template.setConnectionFactory(getRedisFactory());
34         return template;
35     }
36 }
```

4. Create the Controller class and define the apis for VALUE OPERATIONS , LIST OPERATIONS and HASH OPERATIONS .

HOMEWORK :

<1> Change the port of your redis server and start the redis server and client on a new port ?

<2> If we change the port , do we lose out on the keys we have stored as cache ? Will I give the previous port in the config file , will I get my keys back ?

<3> What are the benefits of Redis ?

HINT : What are the benefits of Redis?

- + In-memory data store
- + Flexible data structures : like Strings, Lists, Sets, Sorted Sets, Hashes, Bitmaps, HyperLoglogs
- + Simplicity and ease-of-use
- + Easy for Replication and persistence
- + High Extensibility