图解HashMap

<http://www.jcodecraeer.com/a/anzhuokaifa/androidkaifa/2017/1203/8832.html>

<https://blog.csdn.net/u011958281/article/details/77098013>

<https://blog.csdn.net/sinat_35512245/article/details/60325685>

<https://blog.csdn.net/u010601183/article/details/56496424>

###### Cookies 和 Session的区别

1.cookie 是一种发送到客户浏览器的文本串句柄，并保存在客户机硬盘上，可以用来在某个WEB站点会话间持久的保持数据。

2.session其实指的就是访问者从到达某个特定主页到离开为止的那段时间。 Session其实是利用Cookie进行信息处理的，当用户首先进行了请求后，服务端就在用户浏览器上创建了一个Cookie，当这个Session结束时，其实就是意味着这个Cookie就过期了。  
注：为这个用户创建的Cookie的名称是aspsessionid。这个Cookie的唯一目的就是为每一个用户提供不同的身份认证。

3.cookie和session的共同之处在于：cookie和session都是用来跟踪浏览器用户身份的会话方式。

4.cookie 和session的区别是：cookie数据保存在客户端，session数据保存在服务器端。  
  简单的说，当你登录一个网站的时候，

         如果web服务器端使用的是session，那么所有的数据都保存在服务器上，客户端每次请求服务器的时候会发送当前会话的sessionid，服务器根据当前sessionid判断相应的用户数据标志，以确定用户是否登录或具有某种权限。由于数据是存储在服务器上面，所以你不能伪造，但是如果你能够获取某个登录用户的 sessionid，用特殊的浏览器伪造该用户的请求也是能够成功的。sessionid是服务器和客户端链接时候随机分配的，一般来说是不会有重复，但如果有大量的并发请求，也不是没有重复的可能性.

         如果浏览器使用的是cookie，那么所有的数据都保存在浏览器端，比如你登录以后，服务器设置了cookie用户名，那么当你再次请求服务器的时候，浏览器会将用户名一块发送给服务器，这些变量有一定的特殊标记。服务器会解释为cookie变量，所以只要不关闭浏览器，那么cookie变量一直是有效的，所以能够保证长时间不掉线。如果你能够截获某个用户的 cookie变量，然后伪造一个数据包发送过去，那么服务器还是认为你是合法的。所以，使用 cookie被攻击的可能性比较大。如果设置了的有效时间，那么它会将 cookie保存在客户端的硬盘上，下次再访问该网站的时候，浏览器先检查有没有 cookie，如果有的话，就读取该 cookie，然后发送给服务器。如果你在机器上面保存了某个论坛 cookie，有效期是一年，如果有人入侵你的机器，将你的  cookie拷走，然后放在他的浏览器的目录下面，那么他登录该网站的时候就是用你的的身份登录的。所以 cookie是可以伪造的。当然，伪造的时候需要主意，直接copy    cookie文件到 cookie目录，浏览器是不认的，他有一个index.dat文件，存储了 cookie文件的建立时间，以及是否有修改，所以你必须先要有该网站的 cookie文件，并且要从保证时间上骗过浏览器

5.两个都可以用来存私密的东西，同样也都有有效期的说法,区别在于session是放在服务器上的，过期与否取决于服务期的设定，cookie是存在客户端的，过去与否可以在cookie生成的时候设置进去。

(1)cookie数据存放在客户的浏览器上，session数据放在服务器上  
(2)cookie不是很安全，别人可以分析存放在本地的COOKIE并进行COOKIE欺骗,如果主要考虑到安全应当使用session  
(3)session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能，如果主要考虑到减轻服务器性能方面，应当使用COOKIE  
(4)单个cookie在客户端的限制是3K，就是说一个站点在客户端存放的COOKIE不能3K。  
(5)所以：将登陆信息等重要信息存放为SESSION;其他信息如果需要保留，可以放在COOKIE中

Java基础面试题

1． 自我介绍

2． 讲一件你印象最深刻的事没事怎么解决的

3． JAVA垃圾回收机制怎么工作的

4． AOP是怎么实现的，有几种方式，分别说说

5． TCP/UDP的区别是什么，怎么实现的，区别是什么，分别用在什么地方

6． Map讲一下你对MAP的认识，用过几种MAP集合，分别说说，然后谈一下HsahMap的性能， 主要谈一下哈希冲撞

7． 创造线程的几种方式，Thread和Runnable接口有什么区别

8． 简单说一下你参加的中兴算法大赛

9． 说一下你做的聊天工具，如果遇到服务突然卡顿，怎么解决

10． 讲一下你对于spring框架的了解

11． Mysql数据库，索引是怎么使用的。

12． 你对分布式是否有所了解

13． 你对数据结构了解么

14． 你为什么想转行做软件呢

15． String 和StringBuffer

16． 浅析HashMap与ConcurrentHashMap的线程安全性

17． 堆排序，快速排序

18． 数据库的四大特征，数据库的隔离级别，

19． 线程同步机制用的什么？当初为什么选择这个机制？

20． 互斥和条件变量区别？互斥是阻塞的吗？当初为什么选择阻塞的机制？

21． 消息队列

22． 分布式系统的了解

<https://blog.csdn.net/zhuxinquan61/article/details/71075051>

TransactionDefinition中定义的属性常量如下：

| **Field(属性)** | **Description(描述)** |
| --- | --- |
| ISOLATION\_DEFAULT | 使用底层数据存储的默认隔离级别 |
| ISOLATION\_READ\_COMMITTED | 表示防止脏读;可能会发生不可重复的读取和幻像读取 |
| ISOLATION\_READ\_UNCOMMITTED | 表示可能会发生脏读，不可重复的读取和幻像读取 |
| ISOLATION\_REPEATABLE\_READ | 表示禁止脏读和不可重复读;可以发生幻影读取 |
| ISOLATION\_SERIALIZABLE | 表示可以防止脏读，不可重复的读取和幻像读取 |
| PROPAGATION\_MANDATORY | 支持当前交易;如果不存在当前事务，则抛出异常 |
| **PROPAGATION\_NESTED** | 如果当前事务存在，则在嵌套事务中执行，其行为类似于PROPAGATION\_REQUIRED |
| PROPAGATION\_NEVER | 不支持当前交易;如果当前事务存在，则抛出异常 |
| PROPAGATION\_NOT\_SUPPORTED | 不支持当前交易;而是总是非事务地执行 |
| **PROPAGATION\_REQUIRED** | 支持当前交易;如果不存在，创建一个新的 |
| **PROPAGATION\_REQUIRES\_NEW** | 创建一个新的事务，挂起当前事务（如果存在） |
| PROPAGATION\_SUPPORTS | 支持当前交易;如果不存在，则执行非事务性的 |
| TIMEOUT\_DEFAULT | 使用底层事务系统的默认超时，如果不支持超时，则为none |

**深入理解Spring源码（一）-IOC容器的定位，载入，注册**

<https://www.jianshu.com/p/01d8c57949a6>

### 通过上面的代码分析，总结一下IOC容器初始化的基本步骤：

1.初始化的入口在容器中实现的refresh()中来完成。  
2.对bean定义载入的IOC容器使用的方法是loadBeanDefinitions。

其中大致的过程就是：通过ResourceLoader来完成资源文件位置的定位，DefaultResourceLoader是默认的实现，同时上下文本身就给出了ResourceLoader的实现，可以从类路径，文件系统，URL等方式来定位资源。如果是XmlBeanFactory作为IOC容器，那么需要为他指定bean定义的资源，也就是说bean定义文件时通过抽象成Resource来被IOC容器处理的，容器通过BeanDefinitionReader来完成定义信息的解析和注册。往往使用的是XmlBeanDefinitionReader来解析bean的XML文件定义文件-实际的处理过程是委托给BeanDefinitionParserDelegate来完成的，从而得到bean的定义信息，这些信息在Spring中使用BeanDefinition对象来表示--这个名字可以让我们想到loadBeanDefinition，registerBeanDefinition这些相关方法，他们都是为处理BeanDefinition服务的，容器解析到BeanDefinition以后，需要把他在IOC容器中注册，这由IOC实现BeanDefinitionRegistry接口来实现，注册过程就是IOC容器内部维护一个HashMap来保存得到的BeanDefinition的过程，这个HashMap是IOC容器持有bean信息的场所，以后对bean的操作都是围绕这个hashmap来完成的

3.然后我们就可以通过BeanFactory和ApplicationContext来享受到SpringIOC的服务了。在使用IOC容器的时候，我们注意到除了少量粘贴代码，绝大多数以正确IOC风格编程的应用代码完成不用关心如何到达工厂，因为容器将把这些对象与容器管理的其他对象勾在一起。基本的策略是把工厂放到已知的地方，最好是放在预期的上线文有意义的地方，以及代码将实际访问工厂的地方。Spring本身对声明式载入Web程序的应用程序上下文，并将其存储在ServletContext中的框架来实现。

https://www.imooc.com/article/31807