http://blog.csdn.net/gebitan505/article/details/51614805

网站应用一般使用Session进行登录用户信息的存储及验证，而在移动端使用Token则更加普遍。它们之间并没有太大区别，Token比较像是一个更加精简的自定义的Session。Session的主要功能是保持会话信息，而Token则只用于登录用户的身份鉴权。所以在移动端使用Token会比使用Session更加简易并且有更高的安全性，同时也更加符合RESTful中无状态的定义。

cookie和session及其区别

（1）session是在服务器保存的一个数据结构，用来跟踪用户的状态，这个数据可以保存在集群、数据库、文件中。

（2）cookie是客户端保存用户信息的一种机制，用来记录用户的一些信息，也是实现session的一种方式。

如何进行Session的会话跟踪

传统的3种会话跟踪：Cookie、隐藏表单域和URL重写

（1）Cookie：Cookie是由服务器端生成，发送User-Agent（一般是浏览器），浏览器会将Cookie的key/value保存到某个目录下的文件内，下次请求同一网站时就发送该Cookie给服务器。

Redis是一个Key-Value结构的内存数据库，用它维护User Id和Token的映射表会比传统数据库速度更快，这里使用Spring-Data-Redis封装的TokenManager对Token进行基础操作：

Token交互流程

1. 客户端通过登录请求提交用户名和密码，服务端验证通过后生成一个Token与该用户进行关联，并将Token返回给客户端。
2. 客户端在接下来的请求中都会携带Token，服务端通过解析Token检查登录状态。
3. 当用户退出登录、其他终端登录同一账号（被顶号）、长时间未进行操作时Token会失效，这时用户需要重新登录。

传统身份验证的方法

HTTP是一种无状态的协议， 也就是它并不知道谁访问谁的应用。这里我们把用户看成是客户端，客户端使用用户名还有密码通过了身份验证，不过下回这个客户端再发请求的时候，还得再验证一下。

解决的方法就是，当用户请求登录的时候，如果没有问题，我们在服务端生成一条记录，这个记录可以说明一下登录的用户是谁，然后把这条记录的ID号发送给客户端，客户端收到以后就把这个ID号存在Cookie里，下次这个用户再向服务端发送请求的时候，可以带着这个Cookie ， 这样服务端会验证一下这个Cookie里的信息，看看能不能在服务端找到对应的记录，如果可以，说明用户已经通过了身份验证，就把用户请求的数据返回给客户端。

上面说的就是Session，我们需要在服务端存储为登录的用户生成的Session ， 这些Session可能会存储在内存，磁盘，或者数据库里。我们可能需要在服务端定期的去清理过期的Session。

基于 Token 的身份验证方法

使用基于 Token 的身份验证方法，在服务端不需要存储用户的登录记录。大概的流程是这样的：

1. 客户端使用用户名跟密码请求登录
2. 服务端收到请求，去验证用户名与密码
3. 验证成功后，服务端会签发一个 Token，再把这个 Token 发送给客户端
4. 客户端收到 Token 以后可以把它存储起来，比如放在 Cookie 里或者 Local Storage 里
5. 客户端每次向服务端请求资源的时候需要带着服务端签发的 Token
6. 服务端收到请求，然后去验证客户端请求里面带着的 Token，如果验证成功，就向客户端返回请求的数据