Token，令牌。

传统的身份验证？

HTTP是一种无状态的协议，也就是说它不知道是谁在访问他。

当用户登陆的时候，有以下几种方法：一、全cookie，将用户名、密码全部存放在cookie中。二、sessionid，用户第一次请求时带上用户名和密码，到达服务端的时候，在服务器生成一条记录，这个记录里说明了谁登陆了，然后把sessionid发回给客户端（硬盘cookie保存），当再次请求的时候带上这个sessionid，服务器再验证sessionid，看看在服务器这里能不能找到对应信息，如果可以，则说明通过了身份验证。这种session机制，服务器会为用户登录生成session，这些session可能会在内存、磁盘、数据库中。我们可能需要在服务器端去定期清理过期的session。

三、Token的验证方式：不需要在服务器端存储用户的登陆机制。

1. 用户使用用户名和密码请求登陆
2. 服务器收到请求，验证用户名和密码
3. 验证成功后，服务端会签发一个Token，再把Token发送给客户端
4. 客户端收到Token后把他存储起来，比如放在Cookie或者Local Storage
5. 客户端每次想服务端请求资源的时候带着服务端签发的Token
6. 服务端收到请求，验证Token，验证成功的话，就返回客户端请求的数据

实现Token验证的方法有很多，还有一些标准方法，比如JWT，JSON Web Tokens。有三个部分，header，payload（有效载荷），signature（签名）。

中间用.分开，并且都会用Base64编码，真正的Token如下：

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJuaW5naGFvLm5ldCIsImV4cCI6IjE0Mzg5NTU0NDUiLCJuYW1lIjoid2FuZ2hhbyIsImFkbWluIjp0cnVlfQ.SwyHTEx\_RQppr97g4J5lKXtabJecpejuef8AqKYMAJc

Header包括两部分，一部分是Token的类型，另一部分是使用的算法，

{

"typ": "JWT",

"alg": "HS256"

}

Payload里面是Token的具体内容，里面可以有标准字段，也可以有自定义字段，

* iss：Issuer，发行者
* sub：Subject，主题
* aud：Audience，观众
* exp：Expiration time，过期时间
* nbf：Not before
* iat：Issued at，发行时间
* jti：JWT ID

Signature有三个部分，header、payload、secret，先用Base64编码的header.payload，再用加密算法加密一下，加密的时候再放入一个secret，相当于一个密码。然后加密。