## JWT

### 详情参考文章：

<https://baijiahao.baidu.com/s?id=1608021814182894637&wfr=spider&for=pc>

### 简述

#### 全称

***是Java web token的简写。***

#### 数据结构

典型的，一个JWT看起来如下图。

它是一个很长的字符串，字符之间通过"."分隔符分为三个子串。注意JWT对象为一个长字串，各字串之间也没有换行符，此处为了演示需要，下图特意分行并用不同颜色表示了。每一个子串表示了一个功能块，总共有以下三个部分：

JWT的三个部分如下。JWT头、有效载荷和签名，将它们写成一行如下。



##### JWT头

JWT头部分是一个描述JWT元数据的JSON对象，通常如下所示。

{

"alg": "HS256",

"typ": "JWT"

}

在上面的代码中，alg属性表示签名使用的算法，默认为HMAC SHA256（写为HS256）；typ属性表示令牌的类型，JWT令牌统一写为JWT。

最后，使用Base64 URL算法将上述JSON对象转换为字符串保存。

##### 有效载荷

有效载荷部分，是JWT的主体内容部分，也是一个JSON对象，包含需要传递的数据。 JWT指定七个默认字段供选择。

iss：发行人

exp：到期时间

sub：主题

aud：用户

nbf：在此之前不可用

iat：发布时间

jti：JWT ID用于标识该JWT

除以上默认字段外，我们还可以自定义私有字段，如下例：

{

"sub": "1234567890",

"name": "chongchong",

"admin": true

}

请注意，默认情况下JWT是未加密的，任何人都可以解读其内容，因此不要构建隐私信息字段，存放保密信息，以防止信息泄露。

JSON对象也使用Base64 URL算法转换为字符串保存。

##### 签名哈希

签名哈希部分是对上面两部分数据签名，通过指定的算法生成哈希，以确保数据不会被篡改。

首先，需要指定一个密码（secret）。该密码仅仅为保存在服务器中，并且不能向用户公开。然后，使用标头中指定的签名算法（默认情况下为HMAC SHA256）根据以下公式生成签名。

HMACSHA256(base64UrlEncode(header) + "." + base64UrlEncode(payload),

secret)

在计算出签名哈希后，JWT头，有效载荷和签名哈希的三个部分组合成一个字符串，每个部分用"."分隔，就构成整个JWT对象。

#### 算法

**Base64URL算法**

如前所述，JWT头和有效载荷序列化的算法都用到了Base64URL。该算法和常见Base64算法类似，稍有差别。

作为令牌的JWT可以放在URL中（例如api.example/?token=xxx）。 Base64中用的三个字符是"+"，"/"和"="，由于在URL中有特殊含义，因此Base64URL中对他们做了替换："="去掉，"+"用"-"替换，"/"用"\_"替换，这就是Base64URL算法，很简单把。

#### 用法

客户端接收服务器返回的JWT，将其存储在Cookie或localStorage中。

此后，客户端将在与服务器交互中都会带JWT。如果将它存储在Cookie中，就可以自动发送，但是不会跨域，因此一般是将它放入HTTP请求的Header Authorization字段中。

Authorization: Bearer

当跨域时，也可以将JWT被放置于POST请求的数据主体中。

#### 注意

1. JWT的最大缺点是服务器不保存会话状态，所以在使用期间不可能取消令牌或更改令牌的权限。也就是说，一旦JWT签发，在有效期内将会一直有效。
2. 不要把敏感信息放进JWT
3. 为了减少盗用和窃取，JWT不建议使用HTTP协议来传输代码，而是使用加密的HTTPS协议进行传输。