### 跨域认证

用户认证的一般流程：

1）、用户向服务器发送用户名和密码。

2）、服务器验证通过后，在当前对话（session）里面保存相关数据，比如用户角色、登录时间等等。

3）、服务器向用户返回一个 session\_id，写入用户的 Cookie。

4）、用户随后的每一次请求，都会通过 Cookie，将 session\_id 传回服务器。

5）、服务器收到 session\_id，找到前期保存的数据，由此得知用户的身份。

该模式扩展性不好，若需要跨域，例如A和B网站是同一家公司的关联服务，需要在A上登录后在B上能自动登录，就需要以下解决方案：

一种是将session持久化存储，另一种是服务器索性不保存 session 数据了，所有数据都保存在客户端，每次请求都发回服务器。JWT即该种实现。

### JWT原理

服务器认证后生成一个JSON对象，发回给用户，类似如下：

|  |
| --- |
| {  "姓名": "张三",  "角色": "管理员",  "到期时间": "2018年7月1日0点0分"  } |

以后用户和服务器通信都要发回这个JSON对象。

### JWT数据结构

JWT大概像这样：



它是一个很长的字符串，中间用点（.）分隔成三个部分。注意，JWT 内部是没有换行的，这里只是为了便于展示，将它写成了几行。

JWT三部分：

* Header（头部）
* Payload（负载）
* Signature（签名）

写成一行即：Header.Payload.Signature

### Header

Header部分是一个JSON对象，描述JWT的元数据，通常如下：

|  |
| --- |
| {  "alg": "HS256", //签名的算法（algorithm）,默认是 HMAC SHA256（写成 HS256）  "typ": "JWT" // 统一为JWT  } |

最后将JSON对象使用Base64URL算法转成字符串。

### Payload

playload用来存放实际需要传递的数据，JWT规定了7个官方字段：

|  |
| --- |
| iss (issuer)：签发人  exp (expiration time)：过期时间  sub (subject)：主题  aud (audience)：受众  nbf (Not Before)：生效时间  iat (Issued At)：签发时间  jti (JWT ID)：编号 |

除此之外，还可以自定义私有字段。

TIPS: JWT 默认是不加密的，任何人都可以读到，所以不要把秘密信息放在这个部分。

最后，该部分也要使用Base64URL算法转成字符串

### Signature

Signature 部分是对前两部分的签名，防止数据篡改。

首先，需要指定一个密钥（secret）。这个密钥只有服务器才知道，不能泄露给用户。

然后，使用 Header 里面指定的签名算法（默认是 HMAC SHA256），按照下面的公式产生签名。

|  |
| --- |
| HMACSHA256(  base64UrlEncode(header) + "." +  base64UrlEncode(payload),  secret) |

算出签名后，把 Header、Payload、Signature 三个部分拼成一个字符串，每个部分之间用"点"（.）分隔，就可以返回给用户。

### JWT的使用方式

客户端收到服务器返回的 JWT，可以储存在 Cookie 里面，也可以储存在 localStorage。

此后，客户端每次与服务器通信，都要带上这个 JWT。

你可以把它放在 Cookie 里面自动发送，但是这样不能跨域，所以更好的做法是放在 HTTP 请求的头信息Authorization字段里面。

|  |
| --- |
| Authorization: Bearer <token> |

### JWT特点

（1）JWT 默认是不加密，但也是可以加密的。生成原始 Token 以后，可以用密钥再加密一次。

（2）JWT 不加密的情况下，不能将秘密数据写入 JWT。

（3）JWT 不仅可以用于认证，也可以用于交换信息。有效使用 JWT，可以降低服务器查询数据库的次数。

（4）JWT 的最大缺点是，由于服务器不保存 session 状态，因此无法在使用过程中废止某个 token，或者更改 token 的权限。也就是说，一旦 JWT 签发了，在到期之前就会始终有效，除非服务器部署额外的逻辑。

（5）JWT 本身包含了认证信息，一旦泄露，任何人都可以获得该令牌的所有权限。为了减少盗用，JWT 的有效期应该设置得比较短。对于一些比较重要的权限，使用时应该再次对用户进行认证。

（6）为了减少盗用，JWT 不应该使用 HTTP 协议明码传输，要使用 HTTPS 协议传输。