链滴

golang JWT 包生成 Token, 验证 Token

作者：K
原文链接：https://ld246.com/article/1540349739379
来源网站：链滴

简介

- json web token 简称 jwt 是一种轻量级的 规范
- 常用与用户认证
- 大致由3部分构成:
- Header (头部)
- Payload (载荷)
- Signature (签名)
- 用 .拼接
- Token = Header + '.' + Payload + '.' + Signature

Header

用来表明签名的加密算法token类型等.

```
{
    "typ": "JWT",
    "alg": "HS256"lag
}
```

以上json转为 base64 生成 header

Payload

Payload 记录你需要的信息. 其中应该包含 Claims

Claims

```
Audience string `json:"aud,omitempty"`
ExpiresAt int64 `json:"exp,omitempty"`
Id string `json:"jti,omitempty"`
IssuedAt int64 `json:"iat,omitempty"`
Issuer string `json:"iss,omitempty"`
NotBefore int64 `json:"nbf,omitempty"`
Subject string `json:"sub,omitempty"`
```

1. aud 标识token的接收者.
2. exp 过期时间.通常与Unix UTC时间做对比过期后token无效
3. jti 是自定义的id号
4. iat 签名发行时间.
5. iss 是签名的发行者.
6. nbf 这条token信息生效时间.这个值可以不设置,但是设定后,一定要大于当前Unix UTC,否则token会延迟生效.
7. sub 签名面向的用户

Signature

原文链接：golang JWT 包生成 Token, 验证 Token

通过 header 生明的加密方法生成 签名.

JWT 包下载

go get github.com/dgrijalva/jwt-go

简单使用

生成 Token

Payload 结构体

```
type jwtCustomClaims struct {
  jwt.StandardClaims

  // 追加自己需要的信息
  Uid uint `json:"uid"`
  Admin bool `json:"admin"`
}
```

编写生成 token 的函数

```
/**
 * 生成 token
 * SecretKey 是一个 const 常量
 */
func CreateToken(SecretKey []byte, issuer string, Uid uint, isAdmin bool) (tokenString string, e
r error) {
    claims := &jwtCustomClaims{
      jwt.StandardClaims{
          ExpiresAt: int64(time.Now().Add(time.Hour * 72).Unix()),
          Issuer:    issuer,
      },
      Uid,
      isAdmin,
  }
  token := jwt.NewWithClaims(jwt.SigningMethodHS256, claims)
  tokenString, err = token.SignedString(SecretKey)
  return
}
```

test 一下

```
func TestCreateToken(t *testing.T)  {
  token, _ := CreateToken([]byte(SecretKey), "YDQ", 2222, true)
  fmt.Println(token)
}
```

结果是这样的.

```
=== RUN   TestCreateToken

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE1NDA2MDc2MzYsImlzcyI6IllEUSIsInVpZCI
```

```
MjIyMiwiYWRtaW4iOnRydWV9.oaX63ScaDttkhC31bgjvPSr4PjvBb55UanAA_QP5zpc
--- PASS: TestCreateToken (0.00s)
PASS
```

解析 Token

解析函数

```
/**
 * 解析 token
 */
func ParseToken(tokenSrt string, SecretKey []byte) (claims jwt.Claims, err error) {
  var token *jwt.Token
  token, err = jwt.Parse(tokenSrt, func(*jwt.Token) (interface{}, error) {
    return SecretKey, nil
  })
  claims = token.Claims
  return
}
```

合并到一直测试.

```
func TestCreateToken(t *testing.T)  {
  token, _ := CreateToken([]byte(SecretKey), "YDQ", 2222, true)
  fmt.Println(token)

  claims, err := ParseToken(token, []byte(SecretKey))
  if nil != err {
    fmt.Println(" err :", err)
  }
  fmt.Println("claims:", claims)
  fmt.Println("claims uid:", claims.(jwt.MapClaims)["uid"])
}
```

结果是这样的.

```
=== RUN   TestCreateToken

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE1NDA2MDgyODgsImlzcyI6IllEUSIsInVpZCI
MjIyMiwiYWRtaW4iOnRydWV9.8wE-_Wx-DHI2GMXJ9KT5JOndst2CCEaUNEIGDy9CUbM
claims: map[exp:1.540608288e+09 iss:YDQ uid:2222 admin:true]
claims uid: 2222
--- PASS: TestCreateToken (0.00s)
PASS
```

这里要注意, 信息parse后与签名信息不一致, 会报 err: signature is invalid

END

2018/10/24

祝各位程序猿 1024 节日快乐.~ yum

原文链接：golang JWT 包生成 Token, 验证 Token