



## Урок JWT

## Начинаем программировать

Используя установленный пакет, реализуем создание токена в файле **main.go**

```
package main

import (
    "fmt"
    "time"

    "github.com/golang-jwt/jwt/v5"
)

func main() {
    const hmacSampleSecret = "super_secret_signature"
    now := time.Now()
    token := jwt.NewWithClaims(jwt.SigningMethodHS256, jwt.MapClaims{
        "name": "user_name",
        "nbf":  now.Add(time.Minute).Unix(),
        "exp":  now.Add(5 * time.Minute).Unix(),
        "iat":  now.Unix(),
    })

    tokenString, err := token.SignedString([]byte(hmacSampleSecret))
    if err != nil {
        panic(err)
    }

    fmt.Println(tokenString)
}
```

Здесь с помощью функции `NewWithClaims` мы создаём новый токен, в котором будем хранить имя пользователя. Обратите внимание на остальные поля:

- **iat** — время создания токена
- **nbf** — время, с которого токен станет валидным
- **exp** — время, с которого токен перестанет быть валидным ("протухнет")

Давайте теперь реализуем валидацию токена и извлечение из него данных

```
package main

import (
    "fmt"
    "log"
    "time"

    "github.com/golang-jwt/jwt/v5"
)

func main() {
    const hmacSampleSecret = "super_secret_signature"
    now := time.Now()
    token := jwt.NewWithClaims(jwt.SigningMethodHS256, jwt.MapClaims{
        "name": "user_name",
        "nbf":  now.Add(time.Minute).Unix(),
        "exp":  now.Add(5 * time.Minute).Unix(),
        "iat":  now.Unix(),
    })
```

```
    })

    tokenString, err := token.SignedString([]byte(hmacSampleSecret))
    if err != nil {
        panic(err)
    }

    fmt.Println("token string:", tokenString)

    tokenFromString, err := jwt.Parse(tokenString, func(token *jwt.Token) (interface{}, error) {
        if _, ok := token.Method.(*jwt.SigningMethodHMAC); !ok {
            panic(fmt.Errorf("unexpected signing method: %v", token.Header["alg"]))
        }

        return []byte(hmacSampleSecret), nil
    })

    if err != nil {
        log.Fatal(err)
    }

    if claims, ok := tokenFromString.Claims.(jwt.MapClaims); ok {
        fmt.Println("user name: ", claims["name"])
    } else {
        panic(err)
    }
}
```

Попробуйте запустить. Вы увидите сообщение: token has invalid claims: token is not valid yet

Действительно: ведь он станет валидным через минуту. Мы указали «nbf»: `now.Add(time.Minute).Unix()`. Поменяем значение nbf на «nbf»: `now.Unix()`.

Теперь всё заработает.

Попробуем поменять значение **exp** на «exp»: `now.Unix()`. Теперь при запуске мы увидим token has invalid claims: token is expired. Действительно — ведь наш токен уже истёк (мы сами поставили значение **exp** в **now**)

#### Справка

Исключительное право на учебную программу и все сопутствующие ей учебные материалы, доступные в рамках сервиса, принадлежат АНО ДПО «Образовательные технологии Яндекса». Воспроизведение, копирование, распространение и иное использование программы и материалов допустимо только с предварительного письменного согласия АНО ДПО «Образовательные технологии Яндекса».

[Пользовательское соглашение.](#)

© 2018 – 2024 ООО «Яндекс»