

Token생성

 이해관계자	
 상태	
 작성자	
 문서 유형	
 작성일시	@2022년 7월 11일 오전 12:24
 최종 편집일시	@2022년 7월 11일 오전 12:33
 최종 편집자	

token을 생성하기 위한 jswonwebtoken을 활용

npm install jswonwebtoken —save

```
const jwt = require('jsonwebtoken')
```

```
User.findOne({email:req.body.email} , (err,userInfo) => {
```

-> findOne 은 몽고디비에서 제공하는 함수

```
//요청부
userInfo.comparePassword(req.body.password, (err, isMatch) => {

//모델
userSchema.methods.comparePassword = function(plainPassword, callbackfunc){
  //평문을 암호화해서 db 값과 비교
  console.log("plainPassword ; " + plainPassword);
  console.log("this.password ; " + this.password);
  //로그인시 생성된 토큰값과 비교 모델에 password엔 암호화된 비밀번호가 들어가져 있음
  bcrypt.compare(plainPassword, this.password, function(err, isMatch){
    if(err) return callbackfunc(err)
    return callbackfunc(null, isMatch)
  })
}
```

```
//비밀번호까지 맞다면 USER를 위한 토큰을 생성해야함.
userInfo.generateToken((err, userInfo) =>{
```

```

        if(err){
            return res.status(400).send(err);
        }
        //토큰을 저장한다. 어디에?? 쿠키, 로컬스토리지
        //쿠키에 "x_auth"에 저장됨
        res.cookie("x_auth",userInfo.token)
        .status(200)
        .json({loginSuccess : true, userId: userInfo._id})
    })
})

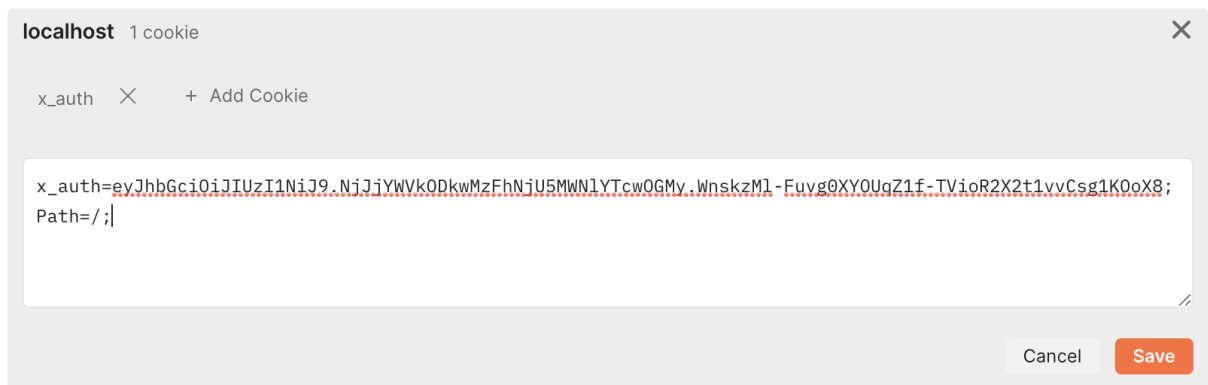
userSchema.methods.generateToken = function(callbackfunc){
    //jsonwebtoken 이용해 토큰 생성
    let user = this;

    //user._id + secretToken이란 txt를 이용해 Token 생성
    //secretToken을 이용해 user._id 도출 가능
    //_id를 통한 토큰값 생성 키는 secretToken
    let token = jwt.sign(user._id.toHexString(), 'secretToken')

    user.token = token
    user.save(function(err, user) {
        if(err) return callbackfunc(err)
        callbackfunc(null, user)
    })
}

```

쿠키값



디비값

QUERY RESULTS: 1-1 OF 1



```
_id: ObjectId("62caed89031a6591cea708c2")
name: "ows"
email: "ows88@naver.com"
password: "$2b$10$6xj fTY1iIz56XsnR0ZL lX.e7m806fpDmwrGnz lfdXHFzHiom6LhT."
__v: 0
token: "eyJhbGciOiJIUzI1NiJ9.NjJjYWVkdDkwMzFhNjU5MmNlYTcwOGMy.WnsKzML-FuvG0XY0..."
```

