Google | 1000 طويق

JWT

أكاديمية طويق
TUWAIQ ACADEMY

وزارة الاتصالات وتقنية المعلومات
MINISTRY OF COMMUNICATIONS AND INFORMATION TECHNOLOGY

الاتحاد السعودي للأمن السيبراني والبرمجة والدرونز
SAUDI FEDERATION FOR CYBERSECURITY, PROGRAMMING & DRONES

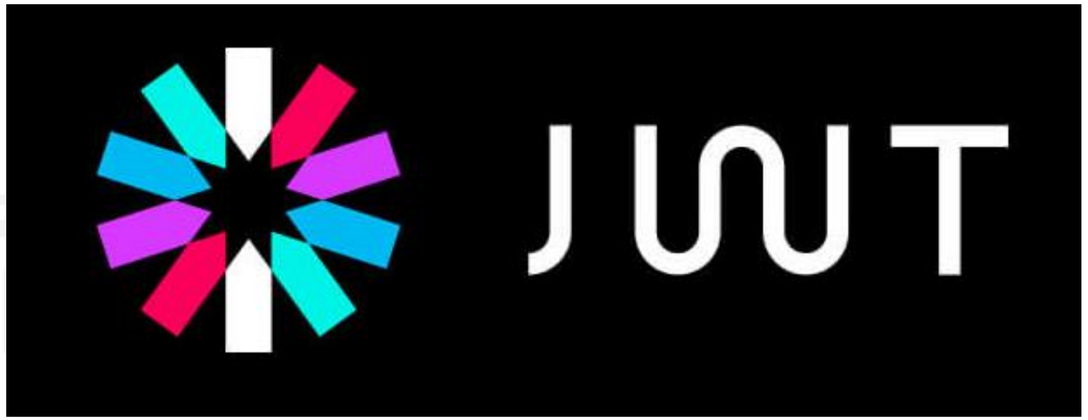
مهارات المستقبل

stc

JWT

- JWT (JSON Web Tokens) to use to authenticate users (login, register)

- Authentication. JWT is mainly used for authentication. After a user logs into an application, the application will create a JWT and send it back to the user. Subsequent requests by the user will include the JWT. The token tells the server what routes, services, and resources the user is allowed to access



JWT Format

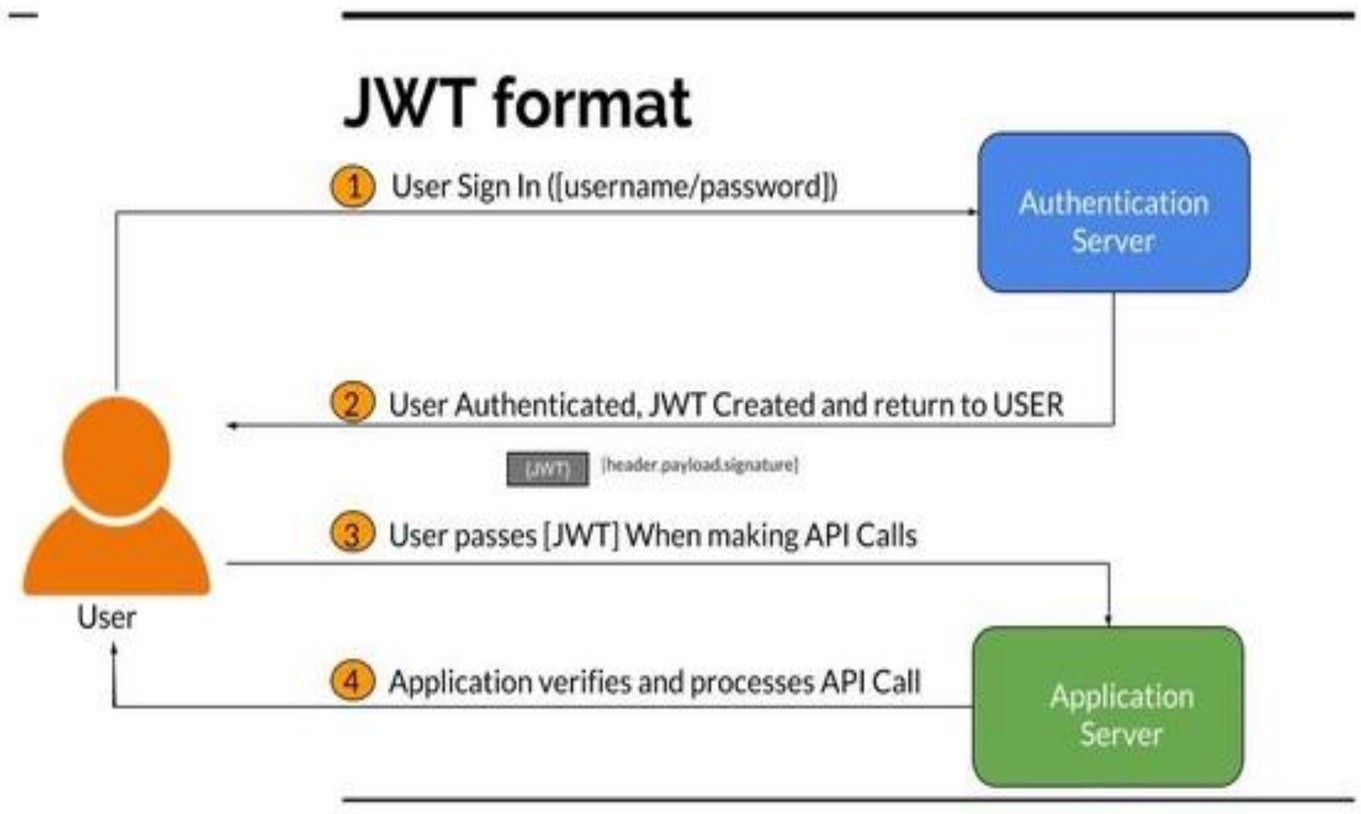
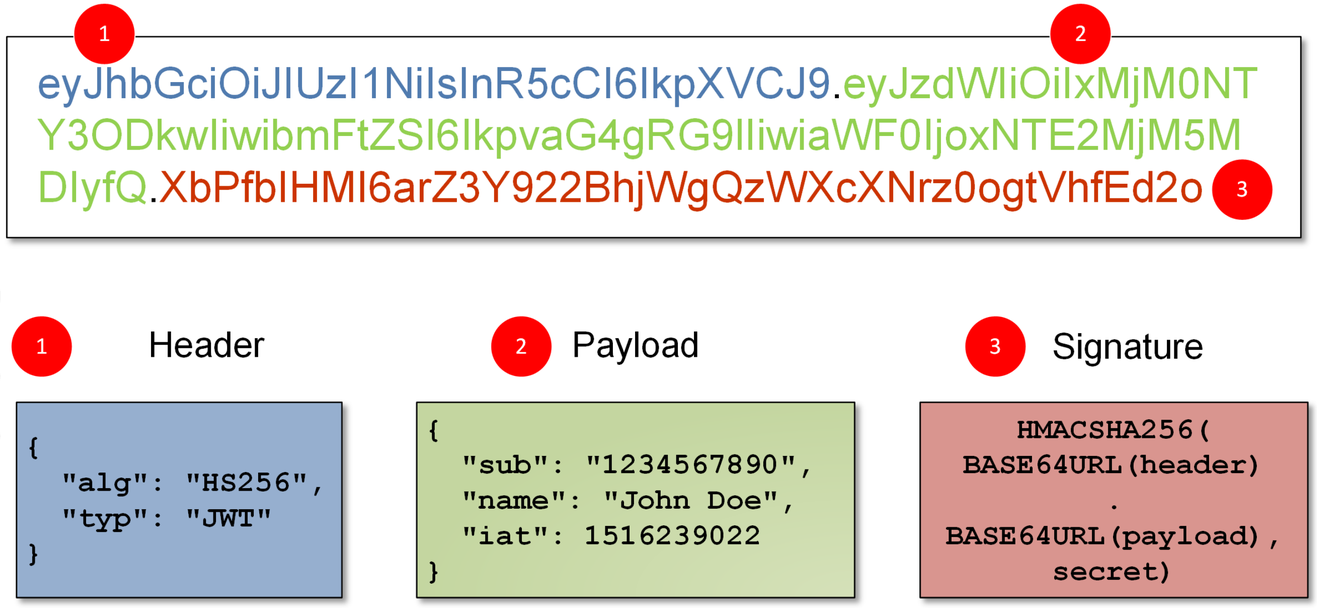


Image source: https://www.mongodb.com/blog/post/the-modern-application-stack-part-2-using-mongodb-with-nodejs

Google | طويق

What is JWT token structure

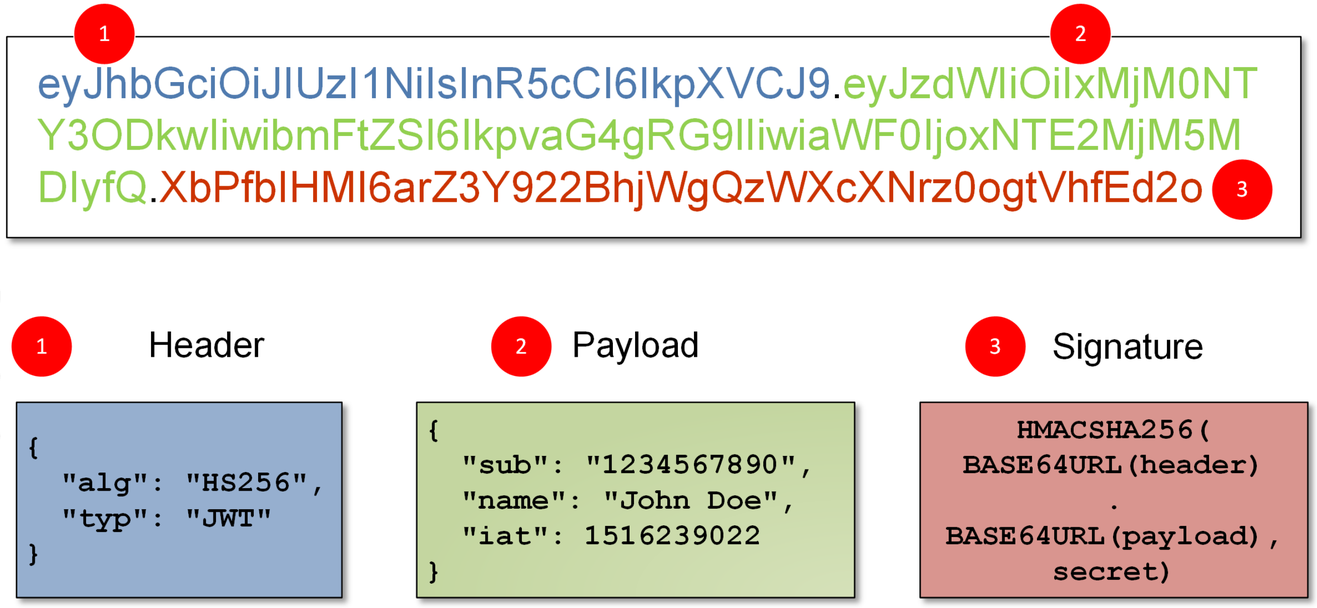
- Header:
  - The header typically consists of two parts: the type of the token, which is JWT, and the signing algorithm being used, such as HMAC, SHA256 or RSA.
- Payload :
  - The second part of the token is the payload, which contains the claims (which has 3 claims), Some of them are: iss (issuer), exp (expiration time), sub (subject)



What is JWT token structure

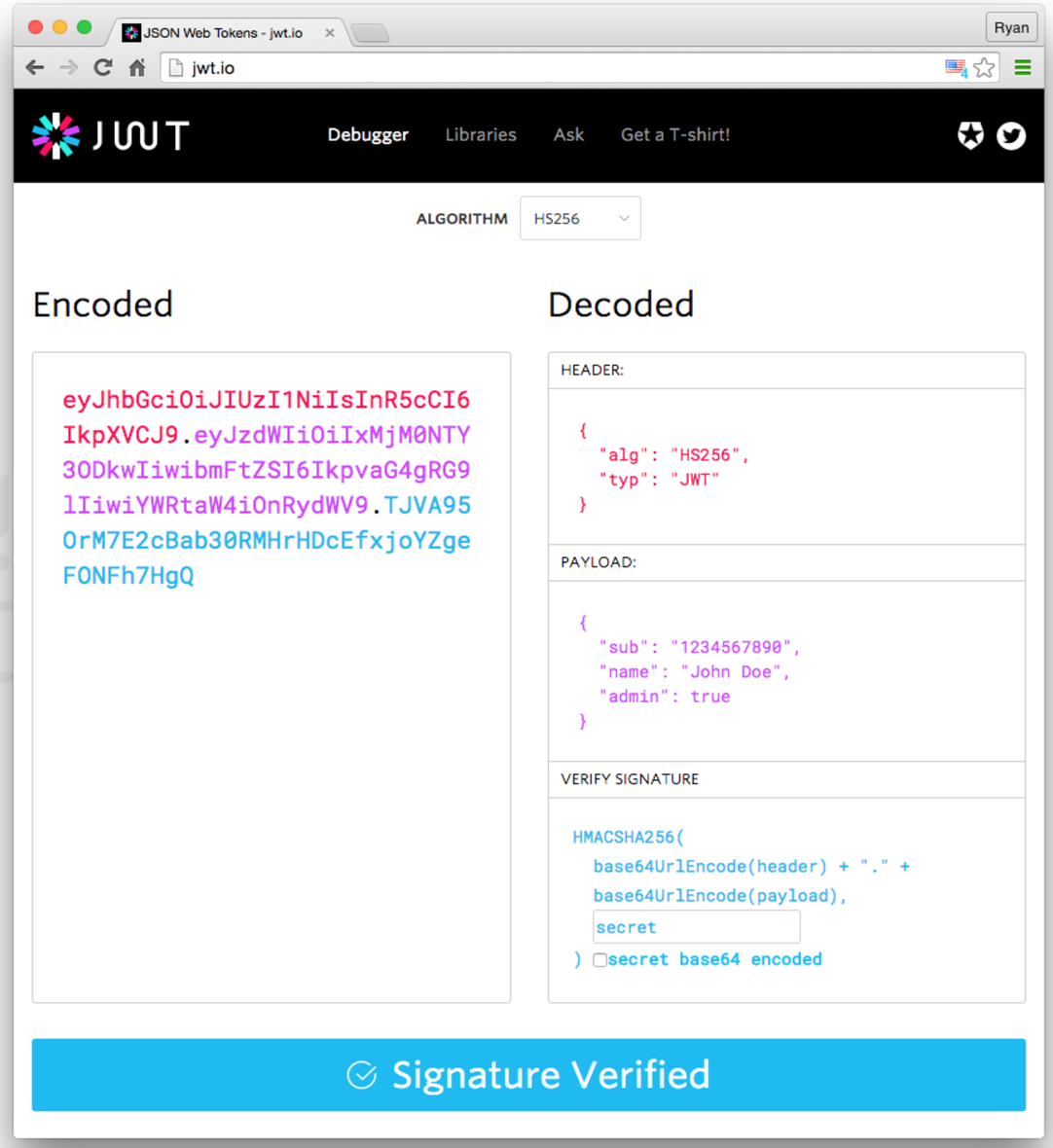
Signature:

- To create the signature part you have to take the encoded header, the encoded payload, a secret, the algorithm specified in the header, and sign that.
- The signature is used to verify the message wasn't changed along the way, and, in the case of tokens signed with a private key, it can also verify that the sender of the JWT is who it says it is.



How the Token Looks

- For more on JWT



Install

- npm I jsonwebtoken

Working with JWT in User Model

- We will need to make sure that we encrypt and secure users when they are logged into the app

```
userSchema.methods.generateAuthToken = async function() {
    const user = this;
    // get current user
    const token  = jwt.sign({_id: user._id.toString()}, 'secret');
    // jwt token has to be the same
    //@@ create the token with the .sign
    user.tokens = user.tokens.concat({ token}) // current user tokens are stored
    await user.save() // save the token to schema
    return token
}

// statics will be model methods
userSchema.statics.findByCredentials = async (email, password) => {
    const user = await User.findOne({email})
    // find the user with the email passed in
    if (!user) {
        throw new Error ('Unable to login')
    }
    const isMatch = await bycrypt.compare(password, user.password);
    // matches the hashed password to log the user in
    if(!isMatch){
        throw new Error ('Unable to login')
        // log here
    }
    return user
    // if there is a match
```

Working with JWT in User Model

Hashing the password for security

```
// Hash the plain password
userSchema.pre('save', async function(next) {
    // before users are saved we will run this method ( we will store passwords here)
    const user = this // store the current user

    if(user.isModified('password')){
        // get current users password and hash it
        user.password = await bycrypt.hash(user.password, 8);
    }
    console.log('Just before saving')
    next() // will save the user when completed
})
const User = mongoose.model('User', userSchema)
module.exports = User;
```

Authentication Middleware

```
const jwt = require('jsonwebtoken');
const User = require('../models/User')
const auth = async (req, res, next) => {
    try {
        // validate user will be here
        const token = req.header('Authorization').replace('Bearer ', '')
        // Header will contain the Auth details
        const decoded = jwt.verify(token, 'secret')
        const user = await User.findOne({_id: decoded._id, 'tokens.token': token});
        if(!user){
            throw new Error('No User is found')
        }
        req.user = user;
        req.token = token;
        // can be used by other functions to be removed
        // If there are no problem
        next()
        console.log(token)
        console.log(decoded)
    }
     catch(e){
        res.status(401).send({error: 'Please authenticate'})
     }
}
module.exports = auth
```

Resources

- Node.js API Authentication With JWT: https://www.youtube.com/watch?v=7nafaH9SddU
- What are Json? : https://www.freecodecamp.org/news/what-are-json-web-tokens-jwt-auth-tutorial/
- The net ninja: https://www.youtube.com/watch?v=SnoAwLP1a-0&list=PL4cUxeGkcC9iqqESP8335DA5cRFp8loyp