**JWT Authentication**

* JSON Web Tokens are an open, industry standard [**RFC 7519**](https://tools.ietf.org/html/rfc7519) method for representing claims securely between two parties.
* JWT.IO allows you to decode, verify and generate JWT
* JWTs are used **as a secure way to authenticate users and share information**.
* Typically, a private key, or secret, is used by the issuer to sign the JWT. The receiver of the JWT will verify the signature to ensure that the token hasn't been altered after it was signed by the issuer
* Language independent

**JWT parts**

* JWT Header
* JWT Payload / Data
* JWT Signature

**JWT Install**

* To use JWT in your project irst install it   
  npm install jsonwebtoken

**Create Token**

* First require jwt in your system
* Then put a secret to your .env file for jwt sign
* Then create a sign with JWT   
  const jwt = require(‘jsonwebtoken’);  
    
  jwt.sign({ id, name, email }, secretKey , {   
   expiresIn : “100d”  
  });
* To create a strong base 64 code for secret key   
  openssl rand -base64 32

**Decode JWT Token**

* After set jwt to your apps, we need it to verify user is logged in or out
* To check this token we need to verify it

const decode\_jwt = jwt.verify(token, secret );

**JWT Expire**

* 100 : 100 milliseconds
* 100s : 100 seconds
* 100h : 100 hours
* 7d : 7days

**Password hash**

* Password should he hash for security reason
* To make hash we can use **bcryptjs** package, its to helpful
* **Install bcryptjs**   
  npm install bcryptjs
* Create a salt for hashing   
  const bcrypt = require(‘bcryptjs’);   
  const salt = bcrypt.genSalt(10);
* Hash make   
  const hashPassword = bcrypt.hash(password, slat);
* Hash verify   
  bcrypt.compare(password, hash);

**Cookie Parser**

* We can store any data in cookie memory for security
* To access cookie memory, install cookie parser   
  npm install cookie-parser
* Now use it in express middleware   
  const cookieParser = require(‘cookie-parser’);   
    
  app.use(cookieParser());
* Set any data in cookie   
  res.cookie(‘cookieName’, data, {  
   expires : Date.now() + new time in milliseconds   
  });
* To check cookie data   
  req.cookies