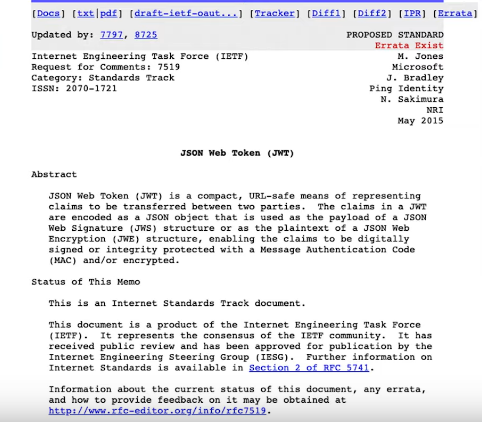
Jwt bortoman shomoye web application jonno khub popular ekta user authorization technique. Especially microservice er khetre eti khub popular. Ei tutorial e amra dekhbo jwt ki, er structure Kemon ebong web application secure korte er vumika ki.

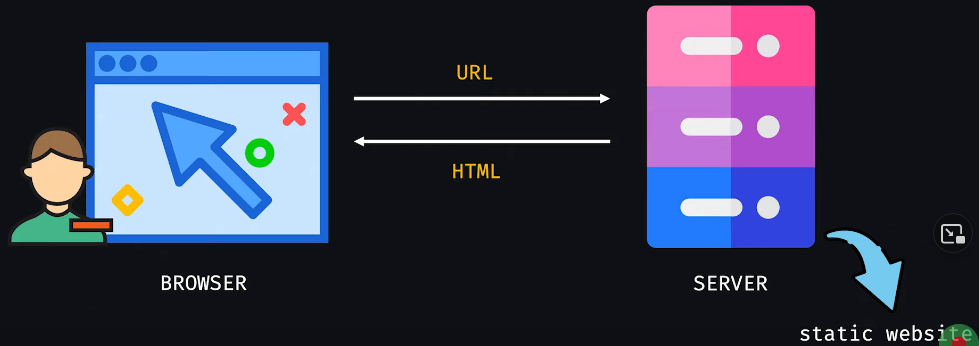
Jwt er er common use case holo eta authorization purpose ei mostly use kora hoy. Kintu jwt creation er main purpose chilo jeno duita party nijeder moddhe ekta standard way te securely communicate korte pare. Industry te RFC 7519 name ekta standard specification ache ja bole dey kivabe jwt structured hobe ebong info exchange korar jonno kivabe eta use korte hobe. Thik jemon ES javascript er standard specification thik kore dey.



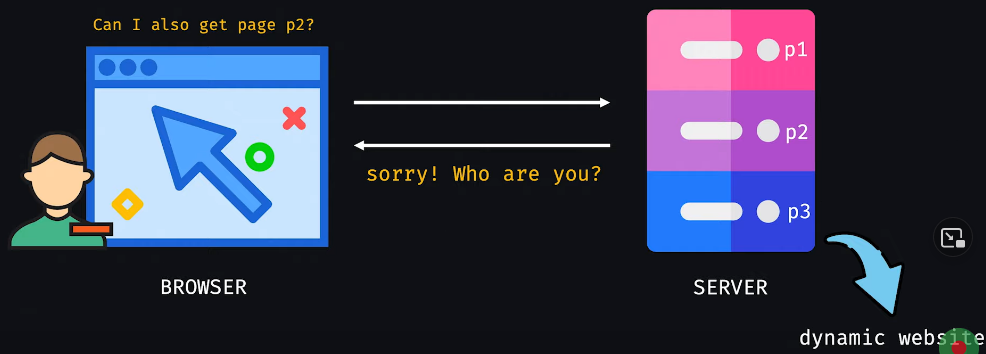
Kintu jehetu jwt mostly authorization purpose e used hoy tai ei tutorial e ami shedikei focus korchi. But mind it, jwt only authorization er jonno create kora hoy nai.

Authorization er jonno jwt charao aro strategy ache jemon ‘session token’. Ei shob strategy (jwt , st etc) er jonmo ekta karone. Sheta holo http er stateless behaviour. http ekta stateless protocol. Amra browser theke server e ba ek server theke arek server e http protocol use kore request , response adan prodan kori. Stateless mane holo client-server interaction er shomoy kono kichu http protocol mone rakhe na. mane prottek ta request korar shomoy request er jonno ja ja proyojon shob kichu pathate hoy. http nije kicchu mone rakhe na. shob vule jay. Ei jonnoi bola hoy http statelss. Beparta ke nicher moto kore vaba jay.

Jokhon amra server theke ekta web page access kori tokhon amra server ke amra ki info dei? Jodi eta ekta simple static website hoy tahole amader beshi kichu pathate hoy na. amra jei pageta dekhte chacchi shetar url pathai ebong server amader corresponding html page ta dey. So server ke ekhtre kono kichu mone rakhte hocche na. kono state maintain korte hocche na. which is perfect for http protocol. Karon shetao stateless.

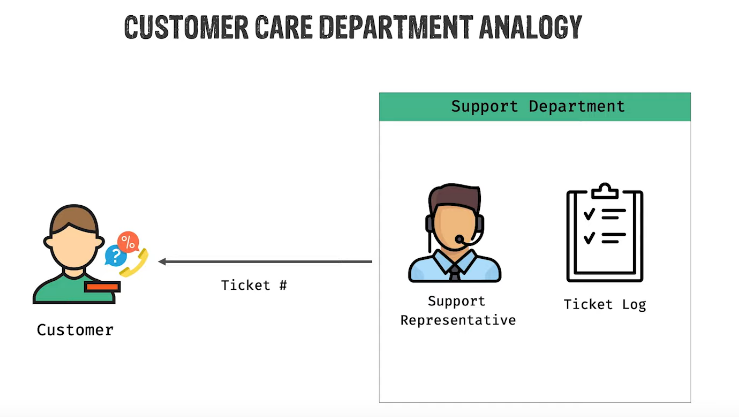


Kintu jodi web app ta emon hoy je ekek user er jonno ekek rokom page response ashbe, mane website ta dynamic ar ki… shekehtre only url pathale hobe na. client ke url ebong nijer identityo pathate hobe. Client page1 er response chaile tahole server ke bolbe “ami user A. amake page1 dao.” Server take page1 dibe. But porer bar jodi she bole “page 2 dao”. Tahole server ki korbe? jehetu http stateless shehetu ebarer request e user er identity na thakay server bujhte parbe na ki response dite hobe.

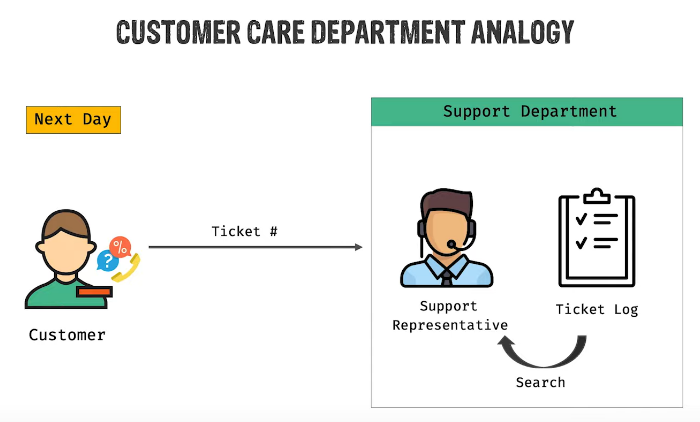


Tai ebareo user ke abar nijer identity diye requested page er response nite hobe. That means jotobar request kora hocche totobar user ke tar identity janan dite hobe. Kintu amra jeshob website e login kore dhuki like facebook --- ekhane logout korar aag porjonto amader erokom bar bar login kora lage na. http jodi stateless hoy tahole eta kivabe possible hocche? Web application kivabe browsing session mone rakhche? web application bivinno vabei session mone rakhte pare. Er moddhe ekta holo token er bebohar. Etar duita popular option ache --- 1. Jwt token 2. Session token.

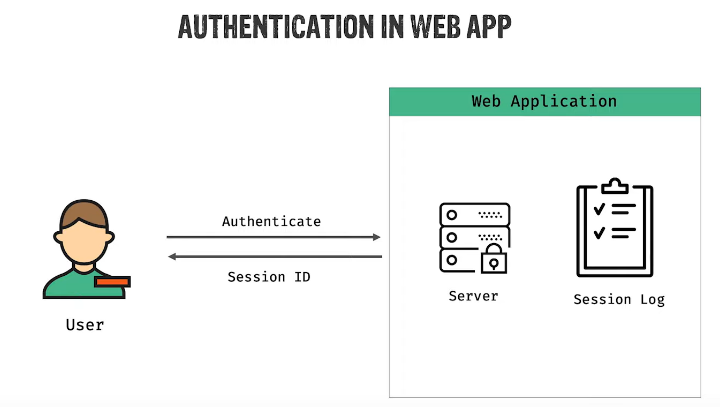
Ekhon amra ekta company customer care department er scenario chinta kori. Shekahne ekjon customer support department e call korlo. Support Representative (SR) customer er complain shune onekgulo troubleshooting koreo solution dite parlo na. she tokhon tar boss ke forward korlo ebong customer tar jonno ekta case file create kore shekhane customer er shathe tar shokol conversation, shob troubleshooting er details likhe rakhlo ebong customer ke oi case id ta (ekta ticket id) diye dilo jeno next time customer ta take call dile ei step gulo take abar korte na hoy.



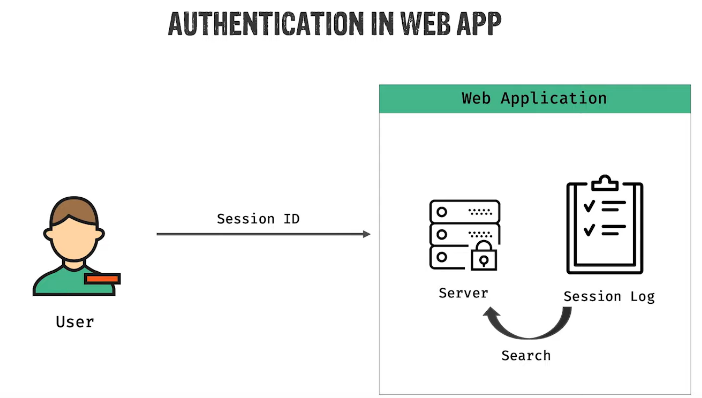
Next din jokhon customer abar call dilo tokhon she tar ticket id ta SR ke dilo. SR system e oi ticket id diye search diye details peye gelo ebong she onujayi customer ke response dilo.



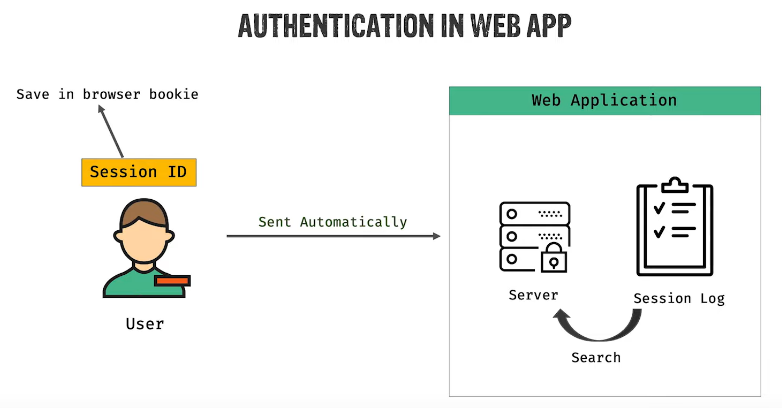
Etokkhon amra je example dekhlam web application e session token diye authentication ta thik evabei hoy. Jokhon user authenticate kore tokhon server ekta session create kore ebong session log e etar ekta track rakhe. Eta oi session er jonno ekta session id create kore ebong user ke dey.



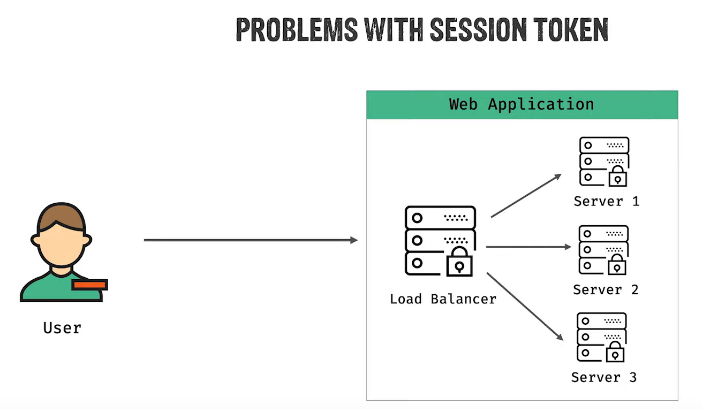
Erpor user jokhoni server e kono request kore tokhon she request er shathe oi session id or token pass kore dey. Server oi session id diye session log e search kore bujhte pare client ta ashole ke. Server k multiple client niye kaj korte hoy. User authenticate korar jonno session token er ei strategy ta besh valo.



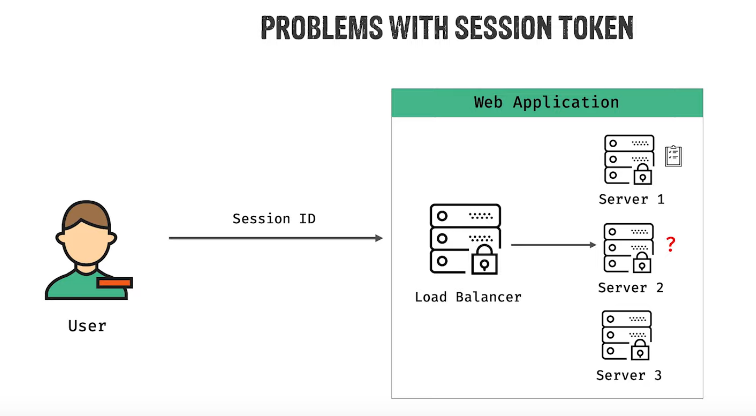
Client kivabe ei session id ta server er kache pass korbe sheta different implementation er upor depend korbe. shobche common approach holo session id ta ke browser er cookie te set kore rakha. Etar shubidha holo jokhoni browser same server e kono request korbe browser automatically ei cookie er info ta request er header e add kore dibe. Eta browser eri ekta achoron. Authenticate korar shomoy server browser er cookie te data save kore dibe. Erpor automatically cookie info protita request e server er kache jete thakbe jeno server client ke chinte pare.



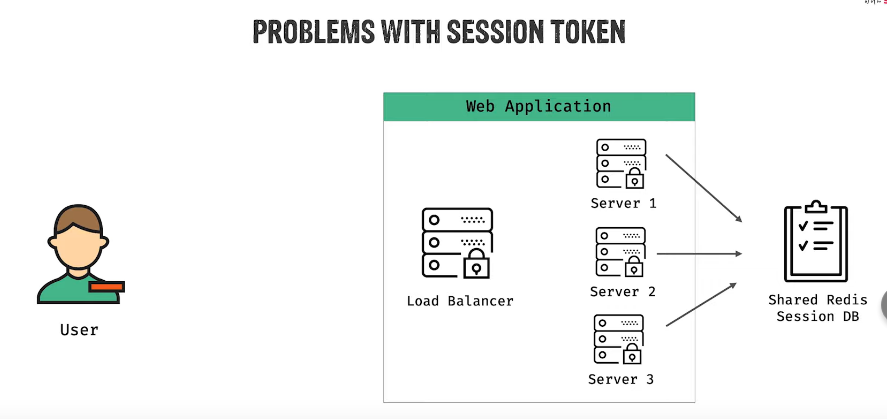
Old web application er khetre eta khubi popular approach chilo. But ei mechanism e besh kichu drawback ache. Ekta problem holo ei mechanism ekta jinish ke chironton shotto dhore niche. She dhorei nicche amader ektai server thakbe. kintu modern web application er khetre multiple server thake. Ekhetre shamner dike ekta load balancer thake. Shei load balancer decision ney user er request ke she kon server e pathabe.



Suppose client session token approach e request korlo. Load balancer request ta server 1 ke forward korlo. Server 1 ekta ekta session id create kore client ke pathay dilo. erpor client session id shoho arekta request korlo. Ebar load balancer server 2 te request ta forward korlo. Kintu server 2 te to session id ta store kora nai. Tahole server 2 kivabe bujhbe eta kon user.



Etar common solution holo, session id gulo specific server e na rekhe ekta common shared redis database e rakha hoy jeno jekono server shekhan theke session id check korte pare. Etake also ‘redis cache’ bola hoy.

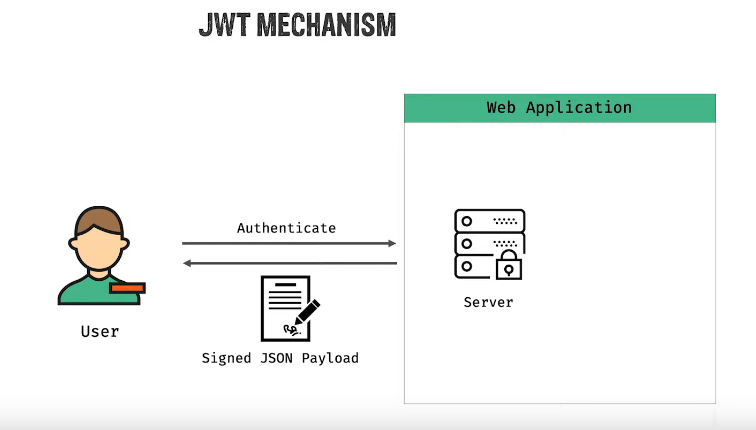


But microservice architecture er khetre ei process er ekta flaw ache. Sheta holo jodi ei redis cache ta kono karone down hoye jay tahole shob server running thaka shotteo authentication mechanism fail korbe. ebong ei problem solve kortei jwt ashche.

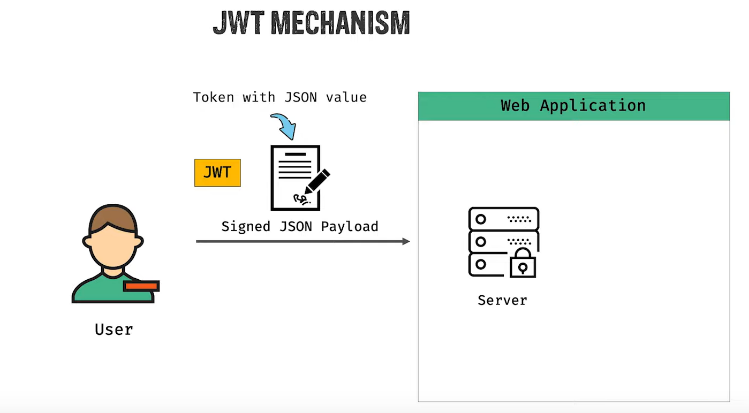
Amader ager customer care department er example ta abar dekhi. Dhori, ebar ar kono phone er system nai. Customer office e ehse SR er shathe dekha korlo. ebar SR ager moto shob info system e store kore na rekhe ekta paper e shob info likhe customer ke diye dilo. ebong customer ke bole dilo next time ashle kagoj ta shathe kore niye ashben. Ekahneo ekta problem ache. Sheta holo validity. Customer jodi valid customer na hoye malicious customer hoy? SR trust korbe kivabe? Porer din jodi customer ekta shada kagoje shob kichu eki vabe likhe ene SR ke dey tahole kivabe SR customer er identity, validity check korbe?

Ekhetre solution holo SR in the first place, paper ta deyar shomoy sign kore dite pare. Erpor jokhon customer sheta niye ashbe tokhon SR shei signature ta validate kore nishchinte serve korte parbe. jwt er concept o same.

Ekehtre client jokhon authenticate korbe tokhon server shob info session log e save kore corresponding session id customer ke na pathaya she user er shob infoi json akare token hishebe pathay dey ebong shei shathe sign o kore dey.

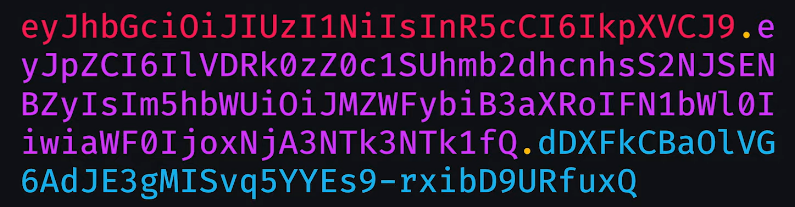


Client er porer request e pura token tai request er shathe pathay dey jekhane lekha ache she kon user, tar porichoy ki etc. server ekhetre kicchu save korche na. shob client er kachei ache. Every time client makes a request with this token server shekhan theke bujhte pare eta kon user ebong tar requested data serve kore. Ei token kono id na. eta ekta json object with all info. It’s called jwt.



Ekhon client ei jwt kivabe save kore rakhbe sheta pura client er responsibility. But ekhetreo most common approach holo browser er cookie or local storage.

Orthat client ke request er shathe json object ta pathate hobe. Kintu jwt dekhte normal json er moto na. eta dekhte erokom.

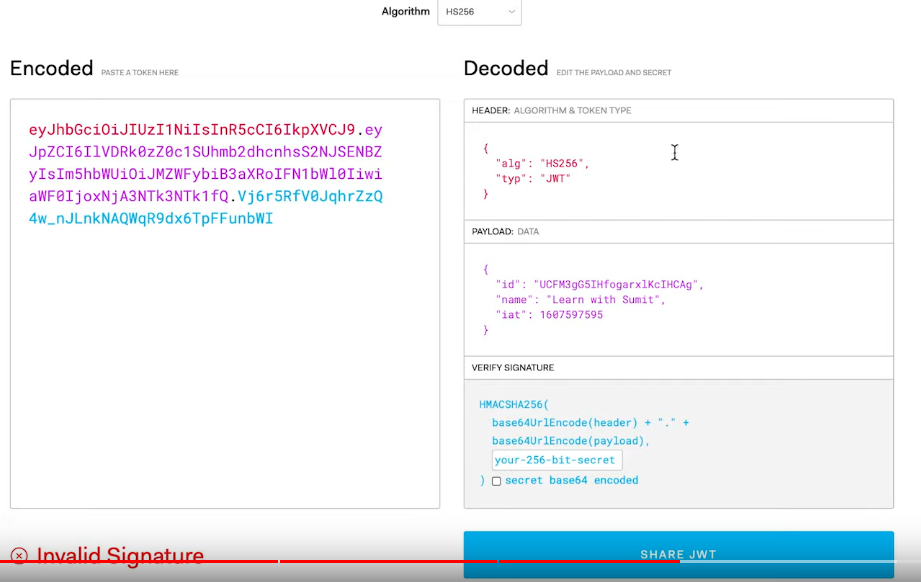


Eta json object eri ekta crypto version. Etar 3 ta part dot(.) diye alada kora ache. Jegulake red, violet, blue color diye mark korsi. 1st part ta holo header part, 2nd ta holo json payload (jeta muloto amar data ta), 3rd ta holo ta holo signature (je signature er kotha ektu age bollam.)

Ekhane header part ta ekta individual json. Payload part tao ekta individual json. 3rd part e signature mane holo simply ekta hashed string. Amader data take ekta secret key diye she hash kore ekta signature toiri kore rakhe. Ei secret key ta server e thake. Jokhoni ei json web token ta server er kache jabe, server tar store kora secret key diye signature ta ke validate korte parbe.

Jwt niye kaj korar jonno ebong er structure bojhar jonno shobche valo website holo jwt.io (<https://jwt.io/>) .

Ektu age je jwt ta dekhlam sheta sumit vai jwt.io website e niye ‘Encoded’ part e paste kore dilo. ei obosthay upore Algorithm chilo ‘HS256’. Paste korar shathe shathe dan pasher ‘Decoded’ section e header, payload ebong verify signature part e kichu lekha dekhlam.

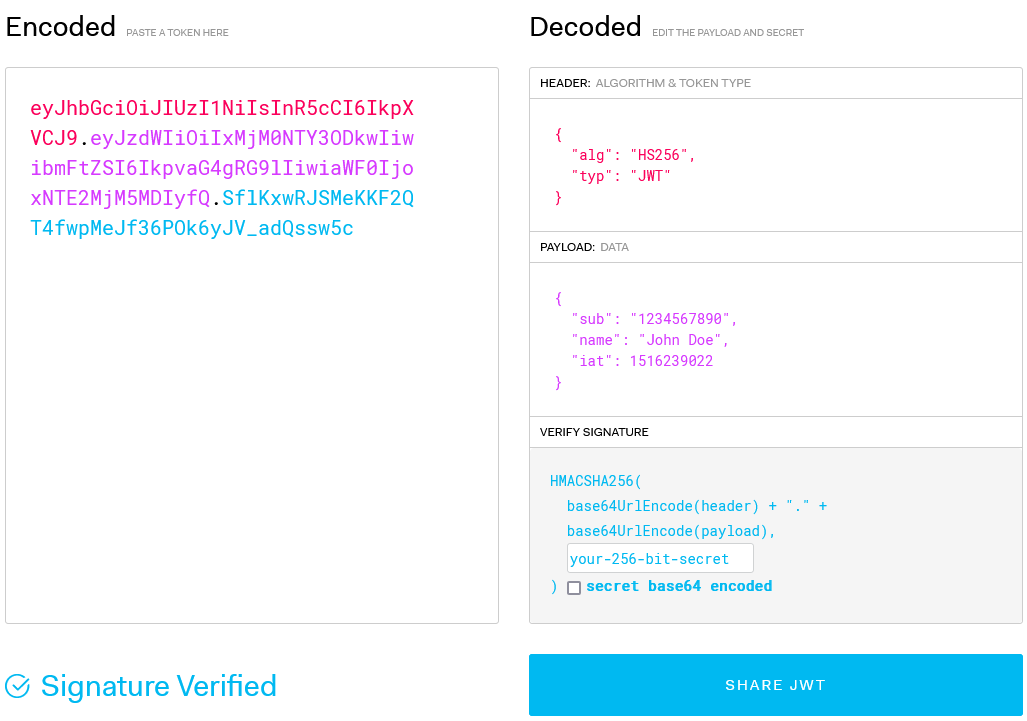


Payload hocche violet color er ongshota. Shetar moddhe self contained data ta ache. Eta ekta json object jetar id, name, iat name 3 ta property ache. Header part eo valid ekta json object ache. Shekhane algorithm ebong type property ache. Eta ashole jwt create korar shomoy je algorithm e eta verify kora hobe shetar meta data ta ekhane ache. main data ta holo payload er json object. Amar 3rd jinish ta holo signature.

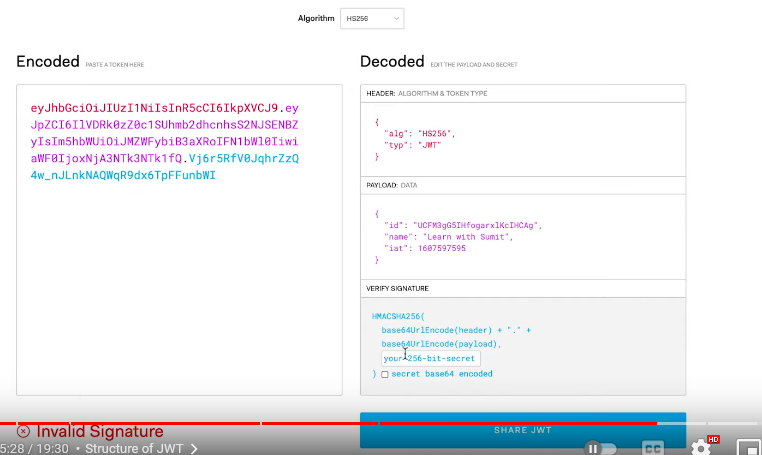
Header ebong payload part e amra normal json object dekhte pacchi. But ‘Encoded’ part e egular corresponding lekhagula hijibiji pacchi keno? Karon normal plain json data tar base64 encoded version hocche hijibiji ta.

Ei hijibiji kintu security purpose e kora hocche na. etake just simplify korr jonno kora hocche. Ei json objectgula to onek borow hote pare. Js niye shobai kaj kore na. onnanno programming language niyeo oneke kaj kore. Oi prog language gulate ei object niye kaj kora problematic hoye jete pare. Tai simplify korar jonno ei hijibiji kora hoise. Tai payload e kono rokomer sensitive data rakha jabe na. only user er public info rakha jabe. Kono password ba secret key rakha jabe na. cz eta easily decode kora jay.

Security ensure kore signature ta. Data er khetre signature deyar system ki? Data er khetre signature deyar system holo eta ke cryptographically secure kore deya with some secret key. Signature ta kivabe toiri hoy?



Header part er base64 encode kora hoy. Tarpor tar shathe concatenate kora hoy payload part er base64 encode. Tarpor je algorithm e create kora hoise shei algorithm e verify korte hobe. Tarpor ekta secret key diye decode kora hobe. Secret key ta public er kache thake na. eta server er kache thake. Server er kache jokhoni hijibiji jwt ta jabe tokhon she header part, payload part gulake evabe kore secret key diye check kore dekhbe eta valid kina.



Sumit vai er ekhane invalid signature dekhacche. She ekhane secret key use korsilo ‘loveyouall’. She ei lekha ta ‘your-256-bit-secret’ field e paste kore dilo. ekhon tar signature verified dekhacche. Jodio she public info dicche but ei jinish ta amader ke validity ensure korche. Json web token kintu password er moto na. it’s kind of user identity jeta dekhe server bujhte pare user valid kina. User er jwt token keu jodi peye jay tahole she login korte parbe.

Session token er khetre jokhoni kono account compromised hoyeche bole mone kora hoy, tokhon session ta ke invalidate kore deya hoy server theke jeno oi session id diye oi session ta te ar keu login na korte pare. But jwt er khetre expiry time shesh na hoye gele eta valid thakbe. tahole jwt ke invalidate korar way ki? Secret key to common shobar jonno. Invalidate korar ekmatro way hocche token blacklist kora. Server e ekta separate databse maintain korte hobe jekhane blacklisted jwt list kora thakbe. jokhoni server e kono user er kach theke request ashbe tokhoni server check kore dekhbe token ta valid kina. Jodi valid hoy tahole arekta middleware er maddhome dekhbe token ta blacklisted kina. Blacklist e na thakle user dhukbe.

Jwt amra bivinno jaygay jemon rest api, microservice er khetre use korte pari. Eta kivabe create korbo sheta programming language er upor depend kore. Jwt ekta universal jinish. Kono nirdishto prog lang er shathe connected na. jwt shudu authentication na, jekono identity verification purpose e use hote pare.