WSA | Forward looking IT finishing school

Authentication and Authorization

Angular

Introduction

(Security Foundations)

Introduction

- **Authentication** is the process of validating a user on the credentials (username and password) and provide access to the web application(ex: Email)

- **Authorization** helps you to control access rights by granting or denying specific permissions to an authenticated user (Ex: User / Manager / Admin).

- Authorization is applied after the user is authenticated. Typically users are assigned with rights / permissions based on which appropriate section(s) are loaded in the web application

- The user interacts with the server on Authorized sections of the application which results in data exchange. In order to protect security and integrity of data other security components (ex: Encryption) comes into picture

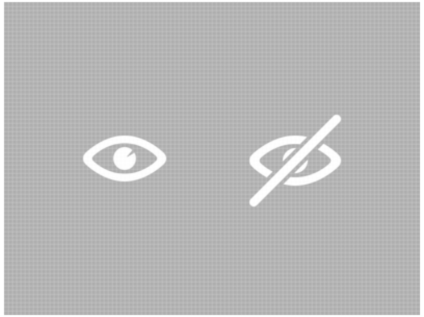
Introduction

- Security is an inherent and critical feature of a web application. With rich data available in the web server, any compromise results in bigger issues in socio / political ecosystem

- There are many algorithms, standards and tools in security which is quite vast in nature

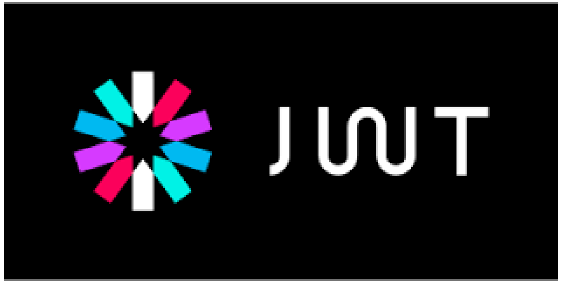
- Our idea is to understand security from Angular Authentication and Authorization perspective by practically implementing them in front-end web applications

- We will enhance our understanding of Routes (previous chapter) and display / hide certain components based on the user authorization

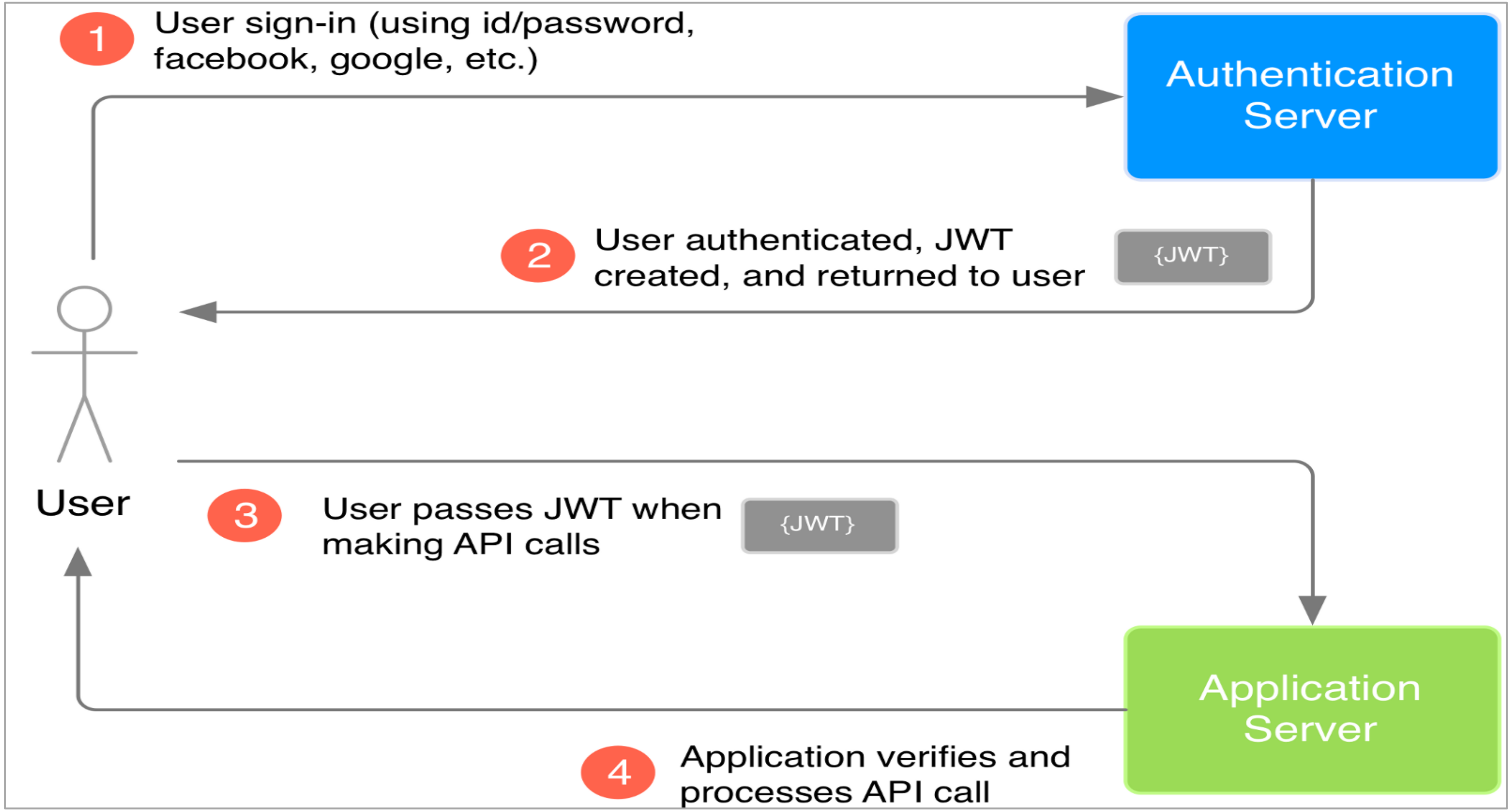


JSON Web Tokens (JWT)

- JSON Web Token (JWT) is an open standard defined in **RFC 7519**.
- It is a compact and self-contained way for securely transmitting information between parties (ex: Web client and server) as a JSON object.
- This information can be verified and trusted because it is digitally signed.
- JWTs are signed using a secret (ex: HMAC algorithm) which is only known to client & server
- The signed token ensures the data integrity and security



JSON Web Tokens (JWT) – In Action..



1 User sign-in (using id/password, facebook, google, etc.)

2 User authenticated, JWT created, and returned to user

3 User passes JWT when making API calls

4 Application verifies and processes API call

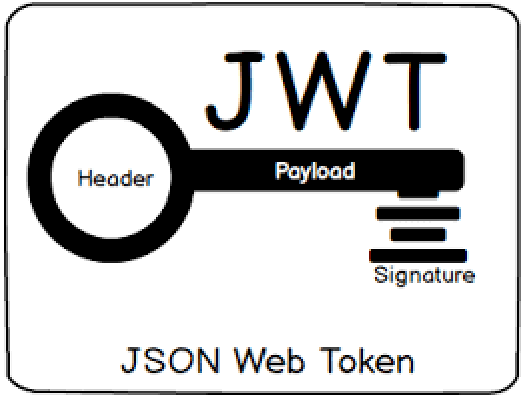
JSON Web Tokens (JWT) – Usage

- JWTs are used in web based authorization once the user is successfully authenticated with valid username & password.

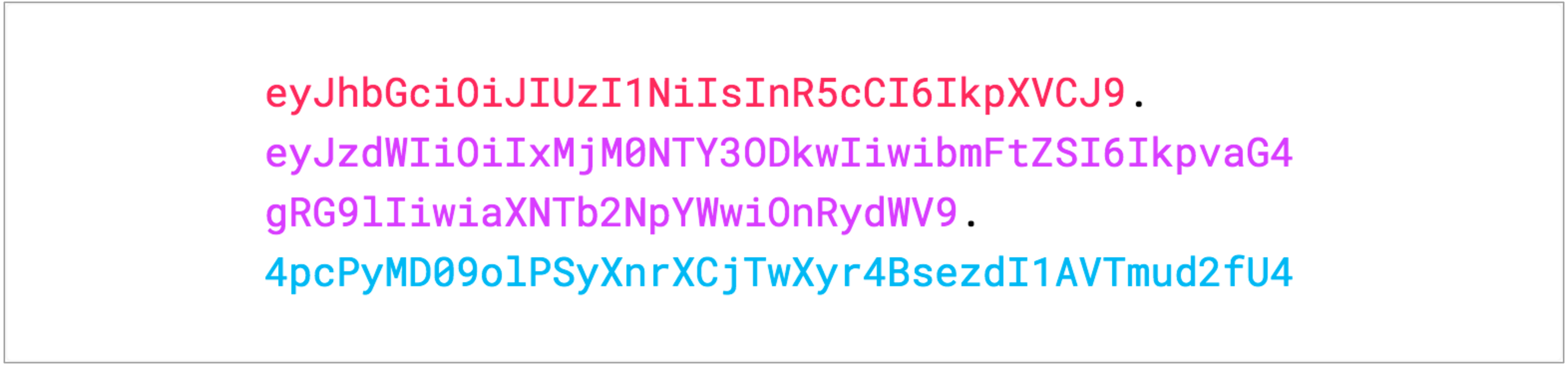
- Each transaction between the client after authorization are done in a secure manner as the data is encrypted.

JSON Web Tokens (JWT) – Structure

- JWT has three parts that are separated by a (.) character
- **Header, Payload and Signature** (ex: xxxx.yyyy.zzzz)
- Each of them have a unique meaning and significance
- An example JWT will look as follows



JSON Web Token



```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4
gRG9lIiwiaXNTb2NpYWwiOnRydWV9.
4pcPyMD09olPSyXnrXCjTwXyr4BsezdI1AVTmud2fU4
```

JWT - Structure

- **Part-I (Header):** Typically consists of two parts:

  - Type of the token (ex: jwt)
  - Hashing algorithm used (ex: HMAC SHA256)

```
{
     "alg": "HS256",
     "typ": "JWT"
}
```

- **Part-II (Payload):** It contains claims. Claims are statements about an entity (typically, the user) and additional data.

- Both Header & Payload are encoded using **base64 encoding** and made as a **first and second part of the JWT**

```
{
   "sub": "1234567890",
   "name": "WSA",
   "admin": true
}
```

JWT - Structure

- **Part-III (Signature):** The signature is nothing but a hash algorithm applied on header and payload

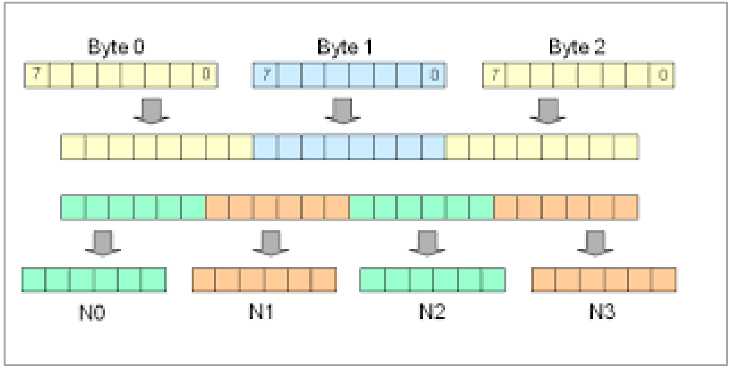
- To create the signature part you have to take the encoded header, the encoded payload, a secret, the algorithm specified in the header, and sign that.

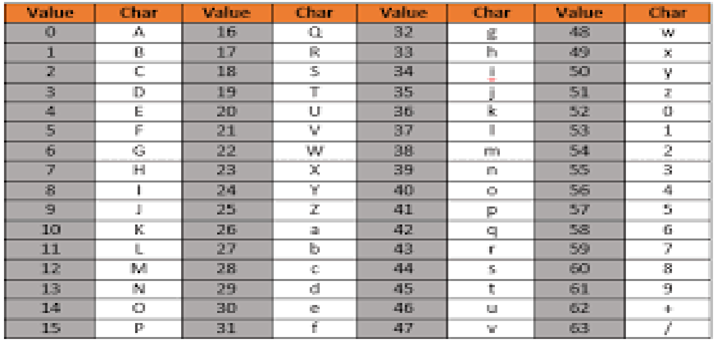
- For example if you want to use the HMAC SHA256 algorithm, the signature will be created in the following way:

```
HMACSHA256 (base64(header) + "." + base64(payload), secret)
```

- The output is three Base64 encoded strings separated by dots that can be easily passed in HTML and HTTP environments

What is base64 Encoding? – A brief





Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

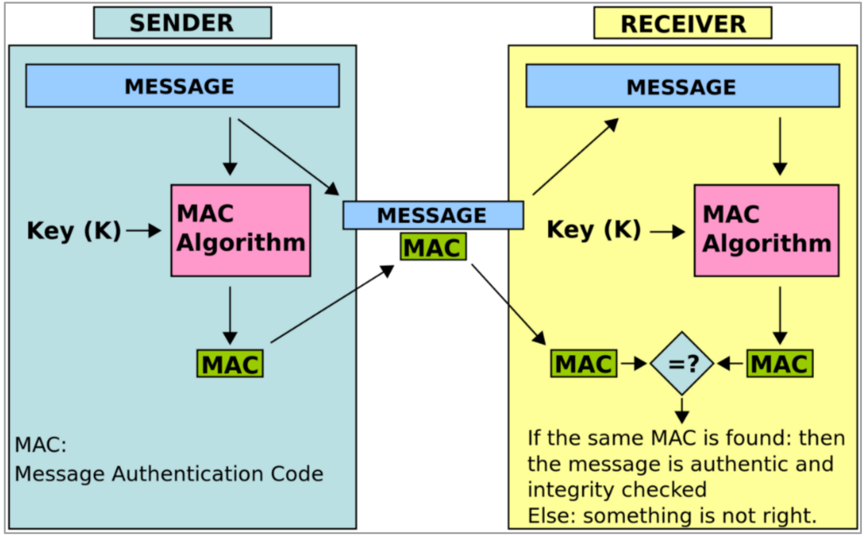
- Base64 converts a string of bytes into a string of ASCII characters so that they can be safely transmitted within HTTP.

- When encoding, Base64 will divide the string of bytes into groups of 6 bits and each group will map to one of 64 characters.

- In case the input is not clearly divisible in 6 bits, additional zeros are added for padding

- Similar to ASCII table a mapping table is maintained

What is HMAC SHA? – A brief



MAC:
Message Authentication Code

If the same MAC is found: then the message is authentic and integrity checked
Else: something is not right.

- HMAC (Hash Message Authentication Code) - SHA (Secure Hash Algorithm) is a specific type of message authentication code (MAC)

- It involves a cryptographic hash function and a secret cryptographic key. The key size can vary (ex: SHA 256)

- The secret key is known only to the sender and the receiver

- By applying hashing it generates what is known as signature of the given plain text. It can be used for validating the integrity of the message.

Exercise

- **JWT Debugger tool:**
  - It is used to generate JWT, let us do some hands-on
  - Goto https://jwt.io/#debugger and try out by generating some JWT

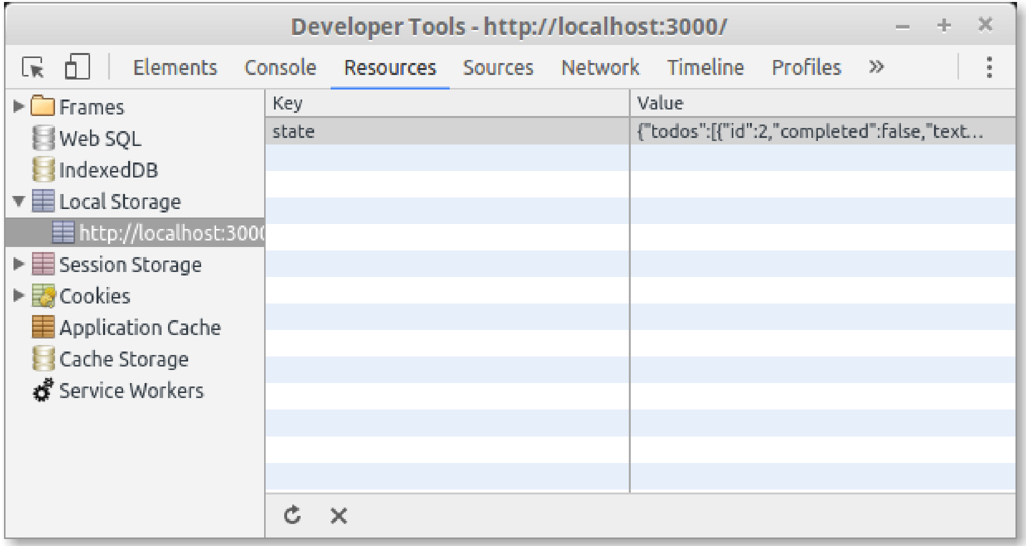
- **Base64 Encoding tool:**
  - It is used to check base64 encoding, let us do some hands-on
  - Goto: https://www.base64decode.org and try out some encoding

WSA | Forward looking IT finishing school

Local Storage

(Storing user data in the browser)

What is Local Storage?



Key	Value
state	{"todos":[{"id":2,"completed":false,"text...

- The Local storage allow to save **key/value pairs** in a web browser.
- The Local storage data will persist after the browser window is closed.
- The local storage property is read-only.
- Previously, cookies were used for storing such key value pairs.
- Local storage has a significantly higher storage limit (**5MB vs 4KB**), better for storing client specific information

Local storage methods

Local storage supports a set of methods for dealing with the data

Method	Description
`setItem()`	Add key and value to local storage
`getItem()`	Retrieve a value by the key
`removeItem()`	Remove an item by key
`clear()`	Clear all storage

Local storage methods usage

```
localStorage.setItem('key', 'value');
localStorage.getItem('key');
localStorage.removeItem('key');
localStorage.clear();
```

Starter Code

(A Brief about given code to get started with A & A)

WSA | Forward looking IT finishing school

Thank you

WebStack Academy

#83, Farah Towers,
1st Floor, MG Road,
Bangalore – 560001

M: +91-809 555 7332
E: training@webstackacademy.com

WSA in Social Media:

slideshare