

JWT

Explained to **5 year old**



Chetan Mahajan
@_chetanmahajan





Without JWT



With JWT

Imagine being at a big music festival with lots of stages and fun things to do.

As soon as you get there, **they give you a special wristband.**

@_chetanmahajan

This wristband is like your key to everything at the festival - **Security guards at each stage simply look at your wristband** to know if you can enter.

They don't need to call the ticket office or check a list.

This is similar to how JSON Web Tokens (JWT) function in the world of web programming.

JWTs are like digital wristbands for online services.



Chetan Mahajan
@_chetanmahajan

@_chetanmahajan

In the digital world, when you log into a website or app, it needs a **way to remember that you're authenticated** (like having a ticket to the festival).

Without JWT, you'd have to log in again every time you switch pages or request data.

That would be like going to the ticket booth every time you want to enter a new stage at the festival - not practical!



Chetan Mahajan
@_chetanmahajan

JWT is a URL-safe, compact string for transferring claims between two parties, made of **three dot-separated parts**:

1. Header (token type and encryption method)
2. Payload (user data and info)
3. Signature (verifies token integrity).

Where it is used?

JWTs are widely used in web applications for user authentication and information exchange.

They're especially popular in Single Page Applications (**SPAs**) and for implementing token-based authentication in **RESTful APIs**.



Chetan Mahajan
@_chetanmahajan

How it's implemented?

When a user logs in, the server issues a JWT. This token is stored by the user's browser and sent back with each request to the server.

Like the festival wristband, it quickly proves the user's identity and access rights, eliminating the need for repeated logins.

Enjoyed this post?

Feel free to **save and share it with others!**



Chetan Mahajan
@_chetanmahajan