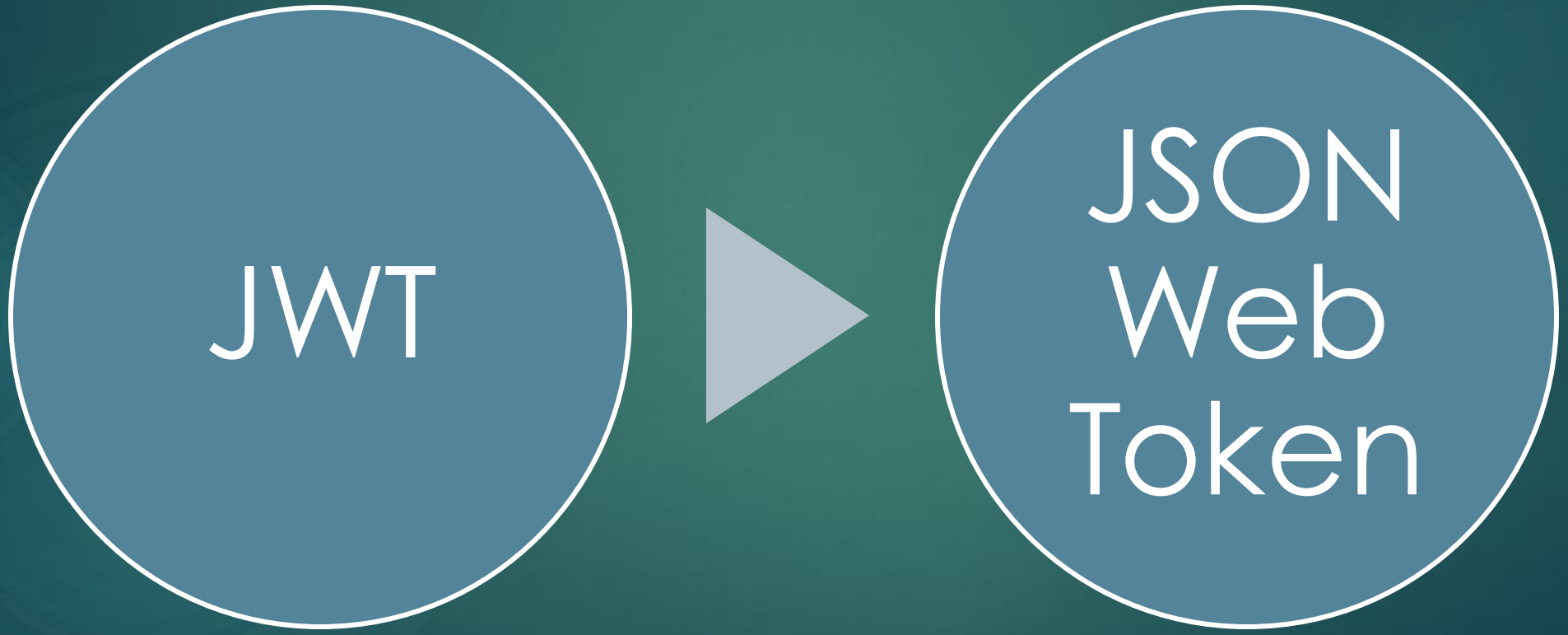
Preeti Rani

ZENSAR TECHNOLOGY


JWT
JSON
Web
Token

WHAT IS JWT?

- In the JWT auth process, the front end (client) firstly sends some credentials to authenticate itself (username and password in our case, since we're working on a web application). The server (the Spring app in our case) then checks those credentials, and if they are valid, it generates a JWT and returns it.

How to implement it using Spring Boot Security?

- Develop a Spring Boot Application to expose a Simple REST GET API with mapping /hello.
- Configure Spring Security for JWT. Expose REST POST API with mapping /authenticate using which User will get a valid JSON Web Token. And then allow the user access to the api /hello only if it has a valid token

Some Important Points of JWT :

- 1. JWT is used for authorization where client passes a header having key as 'Authorization' and value as Bearer<token>.

- 2. JWT is most suitable in SSO(Single sign on) application due to easy usage across different domains . JWT is also used for securely transmitting data across parties.

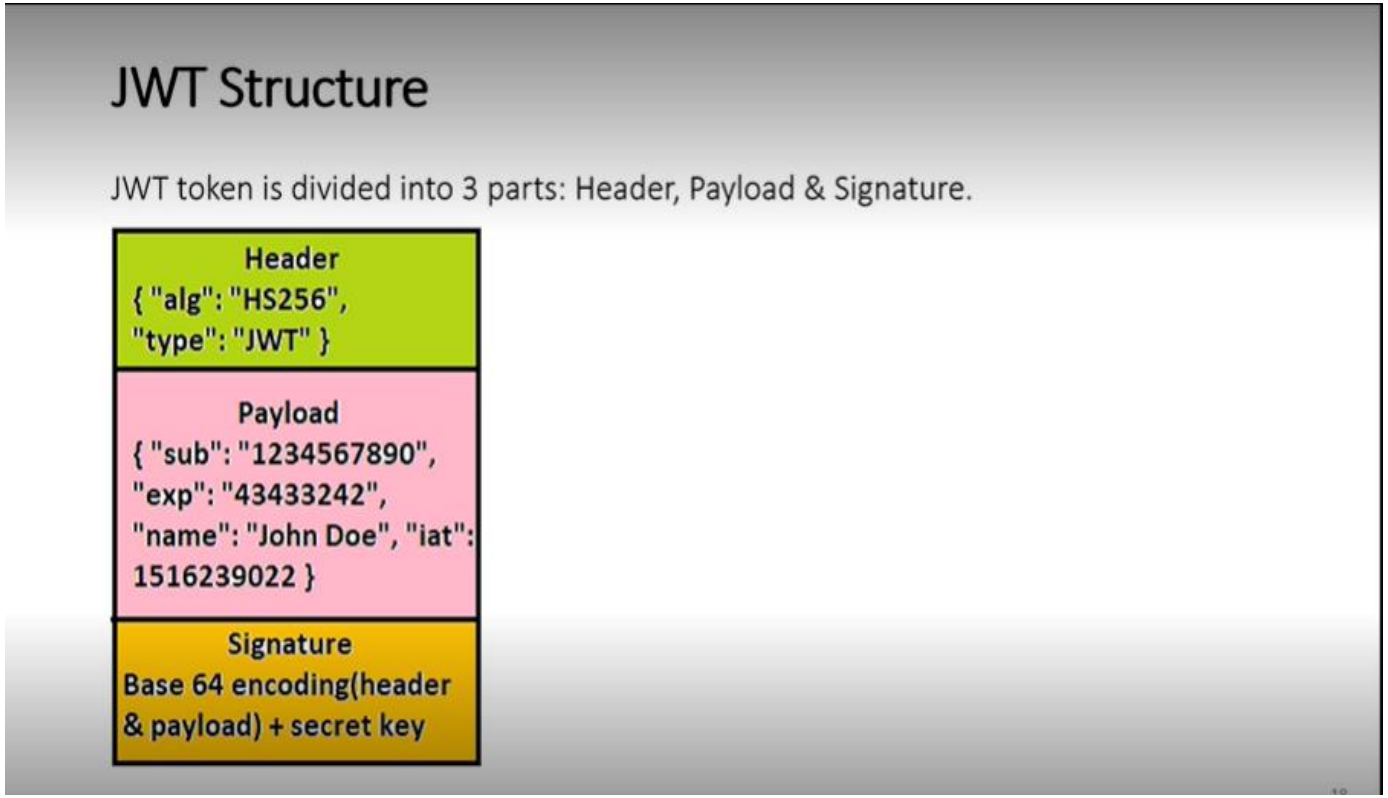
- Suppose in zenlonge appliaction when we navigate the my payroll section the not required to login again because in this case used the JWT token.

- 3. JWT having always three sections and all three section are always encoded section. That is header ,payload and signature.

- 4. Header and payload are decode but signature are not decode.

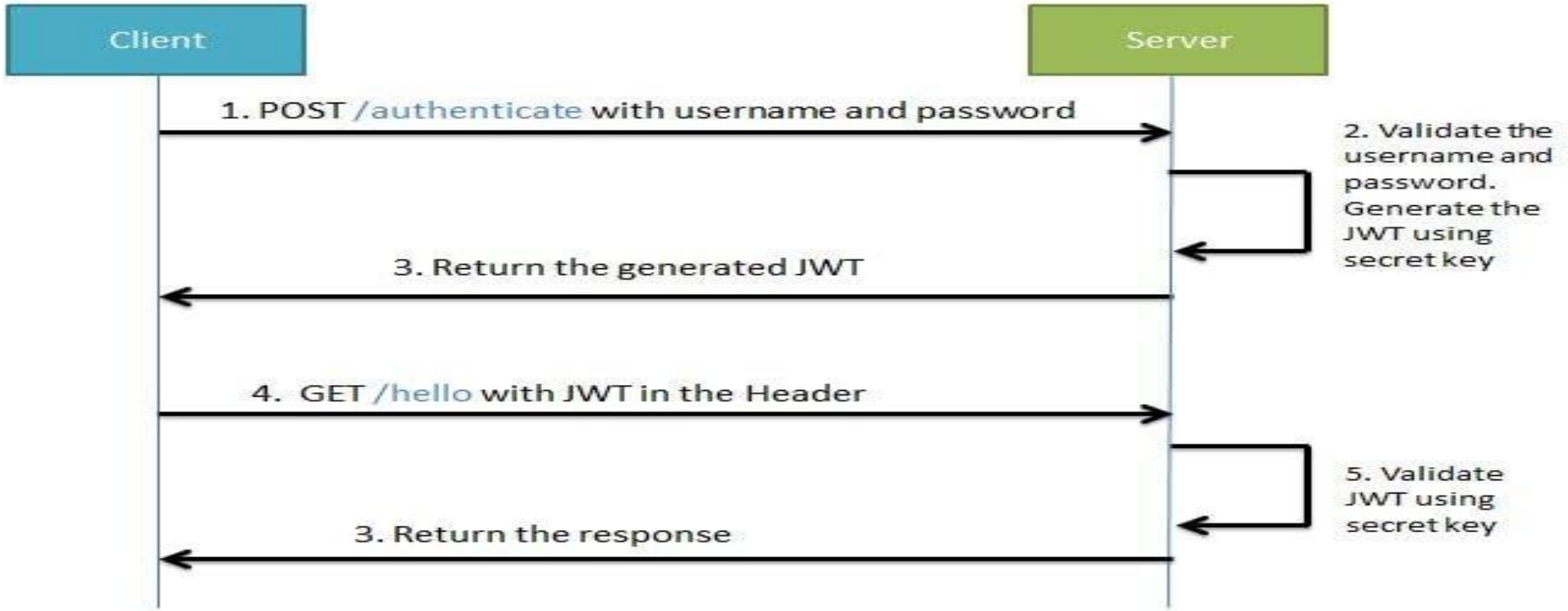
- 4. You can this website 'https://jwt.io/'.

JWT Structure:



Header: Which algoritam are used for encoding.
Signature can't decode because using the secret key.

- Understand using the flow:

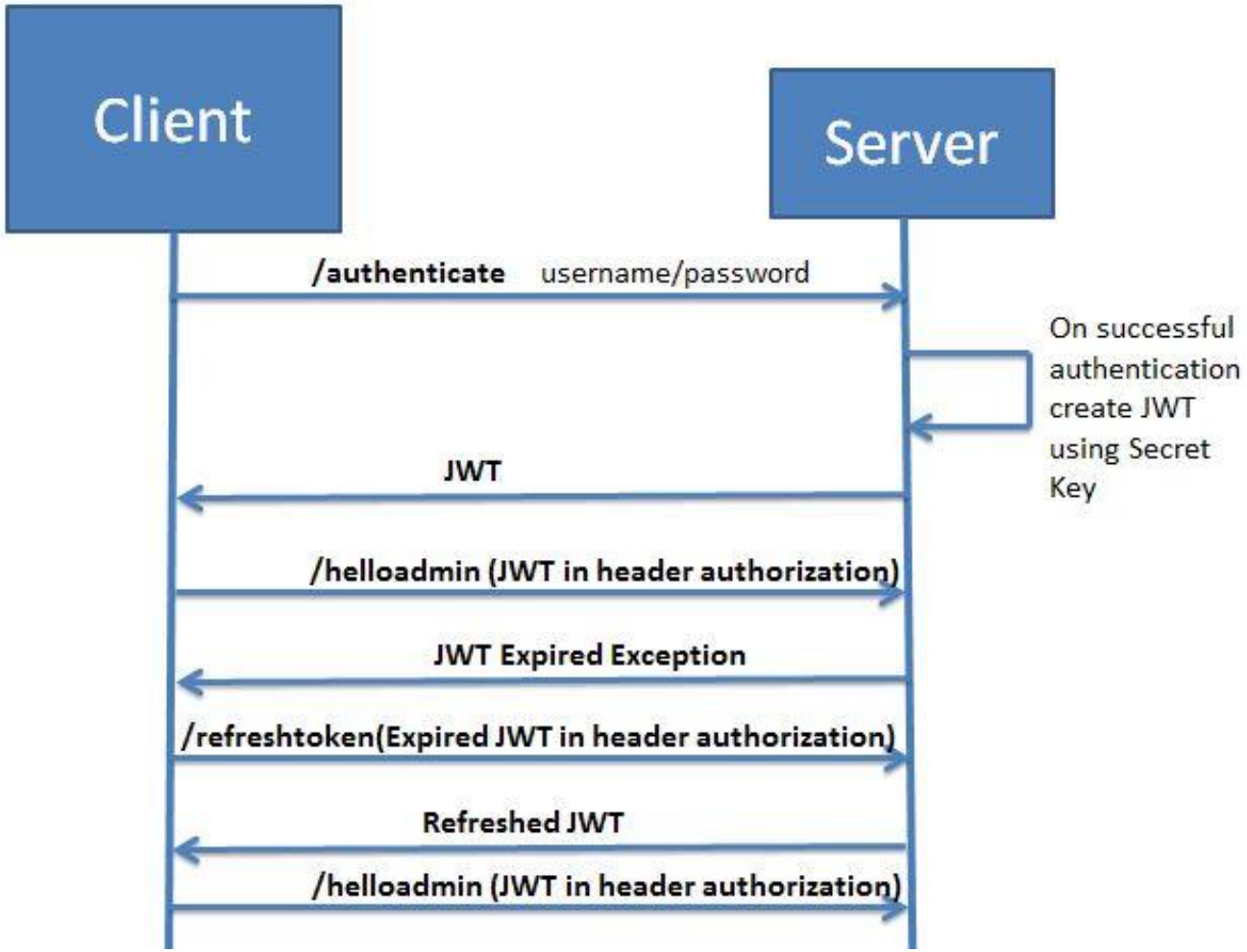


How to implement it using Spring Boot Security?

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-security</artifactId>
    </dependency>
    <dependency>
        <groupId>io.jsonwebtoken</groupId>
        <artifactId>jjwt</artifactId>
        <version>0.9.1</version>
    </dependency>
```

Using Spring Boot Security how to refresh expired JSON Web Token?

- Suppose our requirement is such that if the token has expired, still the user should be allowed to access the system if the token is valid. That is the token should be refreshed or a new valid token should be provided.
We will be working on a solution where if the user he receives JWT expired exception, then he can call another API with the expired token. A new token will then provided to the user which he can use for future interactions. We will be testing this refresh Token generation API both using Postman as well as the Spring RestTemplate.



Understand using this flow:

Advantage of JWT:

- 1. It contains the details of user (not session id in cookies like traditional request), so no need to query database to get user details.

- 2. Can be sent via URL/ Post request/HTTP Header which makes it fast for transmission and usable.

- 3. Information is verified and trusted.

Disadvantage of JWT:

- 1. One of the major cons of relying on tokens is that it relies on just one key. Yes, JWT uses only one key, which if handled poorly by a developer/administrator, would lead to severe consequences that can compromise sensitive information.
- 2. The overall size of a JWT is quite more than that of a normal session token, which makes it longer whenever more data is added to it.
- So, if you're adding more information in the token, it will impact the overall loading speed and thus hamper user experience.
- 3. hort-lived JWT are harder for users to work with. These tokens require frequent reauthorization, which can be annoying at times, especially for the clients.
- Adding refresh tokens and storing them appropriately is the only way to fix this scenario where long-lived refresh tokens can help users stay authorized for a more extended period of time.

Thank You