**Authentication** is the process of identifying users and validating who they claim to be. One of the most common and obvious factors to authenticate identity is a password. If the user name matches the password credential, it means the identity is valid, and the system grants access to the user.

**Authorization** happens after a user’s identity has been successfully authenticated. It is about offering full or partial access rights to resources like database, funds, and other critical information to get the job done.

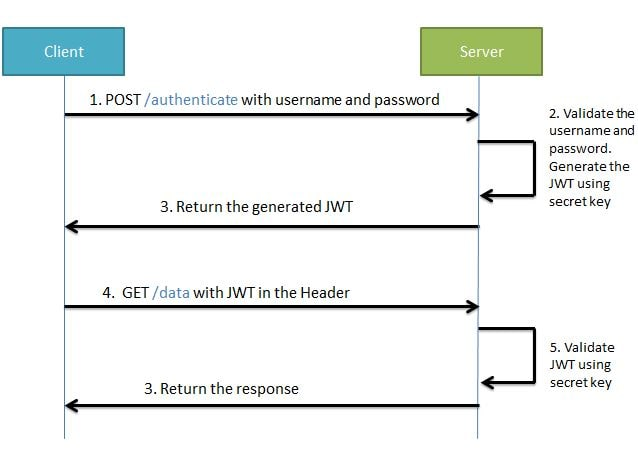
In an organization, for example, after an employee is verified and confirmed via ID and password authentication, the next step would be defining what resources the employee would have access to.

**JWT - Json Web Token**

JWT stands for JSON Web Token. JSON Web Token (JWT) is an open standard that defines a **compact and self-contained way for securely transmitting information between parties as a JSON object**.

This information can be verified and trusted because it is digitally signed. The client will need to authenticate with the server using the credentials only once. During this time the server validates the credentials and returns the client a JSON Web Token(JWT).

For all future requests the client can authenticate itself to the server using this JSON Web Token(JWT) and so does not need to send the credentials like username and password.

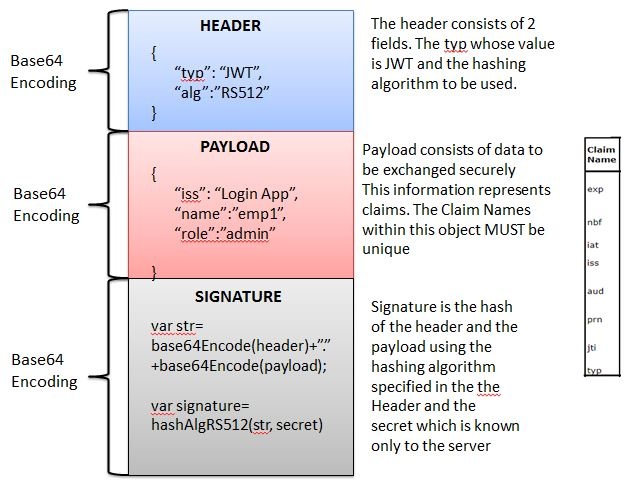


## 

## **Components of JWT**

JWT has the following format -**header.payload.signature**





## 

## **Spring Security and JWT Configuration**

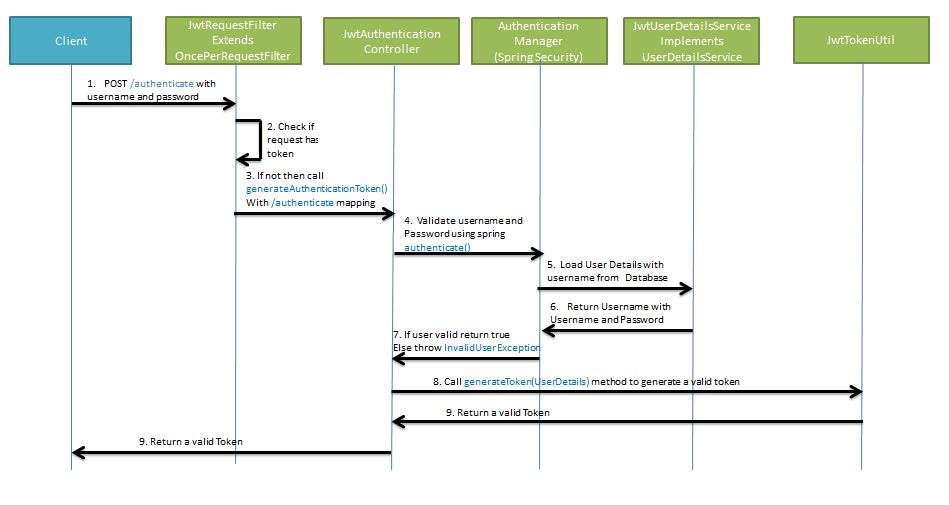
We have to configure Spring Security and JWT and perform below

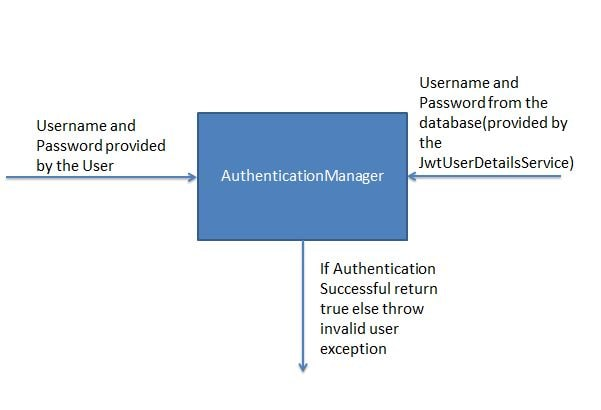
Two operations-

* **Generating JWT** - Expose a POST API with mapping **/authenticate**. On passing correct username and password it will generate a JSON Web Token(JWT)
* **Validating JWT** - If user tries to access GET API with mapping **/welcome**. It will allow access only if request has a valid JSON Web Token(JWT)

The sequence flow for these operations will be as follows-

### **Generating JWT**





### **Validating JWT**

