Learning Consolidation

Authenticate a Backend Application by Using JASON Web Token (JWT)

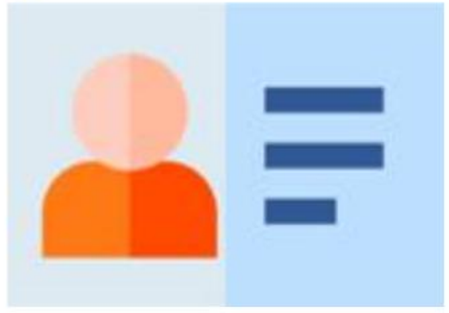
Learning Objectives

- Explain authentication in Spring Boot
- Explore JWT

Authentication and Authorization

Authentication

Identifies who the user is



Authentication is a process that scrutinizes the user at the login level, where the user is verified.

Authorization is a process where each valid user of a system has been given an individual space and resource along with the roles that they possess in a system.

Identifies what permissions does the user has

Authorization

Slide Note

What Is a JSON Web Token?

- The JSON Web Token, or JWT, as it is more commonly called, is an open internet standard for securely and compactly transmitting trusted information between the client and server.
- JWT can be used to authenticate or verify an application user.
- It is a standard for token-based authentications.
- JWT works across different programming languages.
- It can be passed around easily between the client and server.
- The tokens contain claims encoded as JSON objects and digitally signed using a private secret or a public/private key pair.
- They are self-contained and verifiable as they are digitally signed.

Slide Note

The Uses of JSON Web Tokens

- Authorization
  - Once the user is logged in, each subsequent request will include JWT, allowing the user to access controllers, services, and resources that are permitted with that token.
- Information Exchange
  - JSON web tokens are a good way of securely transmitting information between parties.

Slide Note

The user first logs in the application

How Does the JWT Work?

- The user first signs into the application using valid credentials.
- The server authenticates the user and issues a JWT back to the client.
- When the user makes API calls to the application, the client passes the JWT along with the API call.
- The server verifies the validity of the incoming JWT that the client has passed.

Slide Note

JWT
JSON WEB TOKEN

HEADER
ALGORITHM &
TOKEN TYPE

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

PAYLOAD
DATA

```
{
      "sub": "98765",
      "name" : "John"
 }
```

SIGNATURE
VERIFICATION

```
{
   HMACSHA256(
      base64UrlEncode(h
eader) + " . " +
base64UrlEncode(paylo
ad),secretkey) ]}
```

What Does the JWT Look Like?

- The header consists of two parts: the type of token which is JWT, and the signing algorithm being used (HMAC SHA256 in this case).
- Payload - The payload will carry the bulk of JWT, also called the JWT Claims. Claims are statements about an entity (typically the user) and additional data. There are three types of claims: registered, public, and private claims.
- Signature - The third and final part of JWT is the signature. This signature is made up of a hash of the following components:
  - the header
  - the payload
  - secret

Slide Note

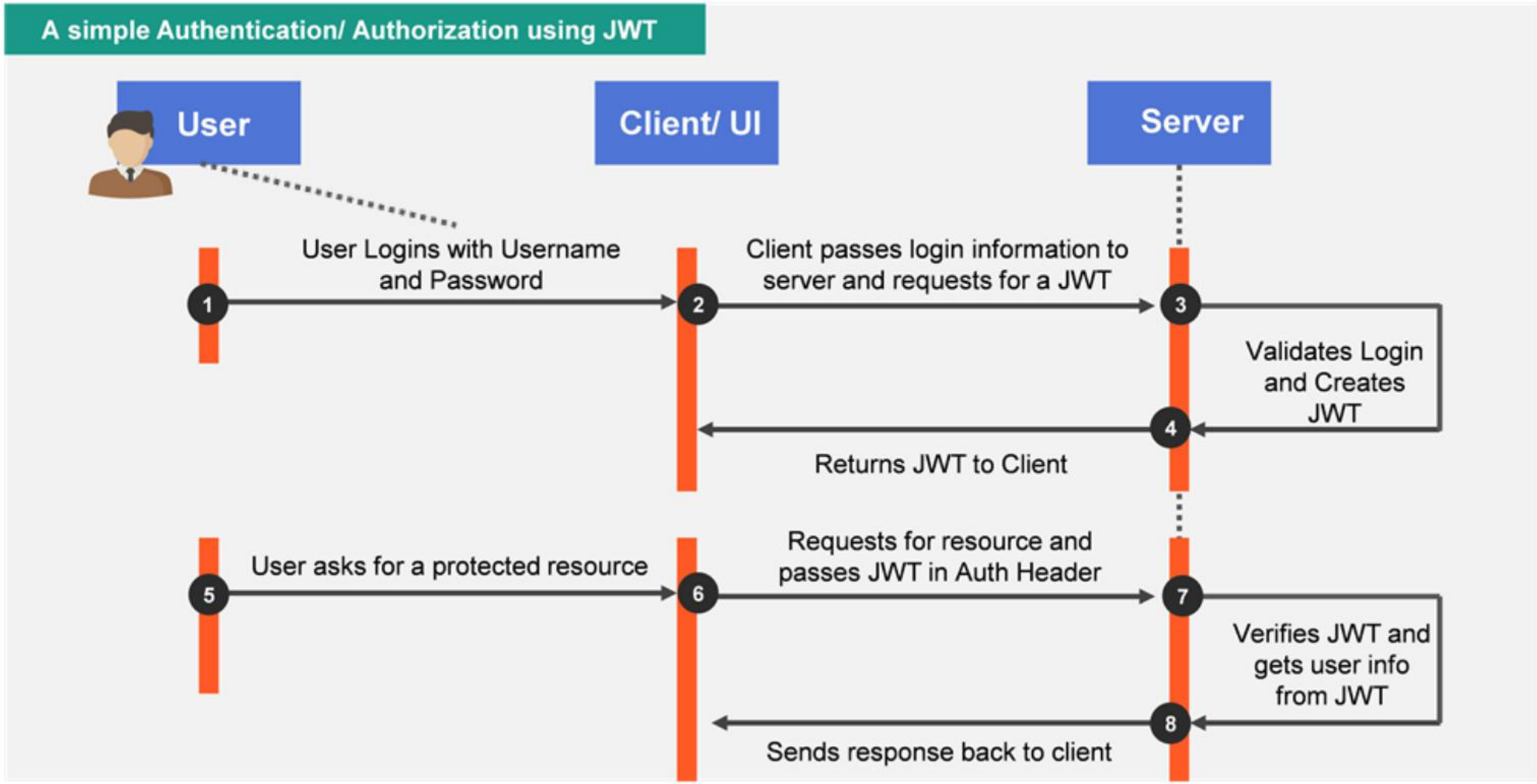
JSON Web Token

- The image below shows a JWT that has a header and payload encoded, and it is signed with a secret key.
- This output is three Base64-URL strings separated by dots that can be easily passed in HTML or HTTP environments.

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

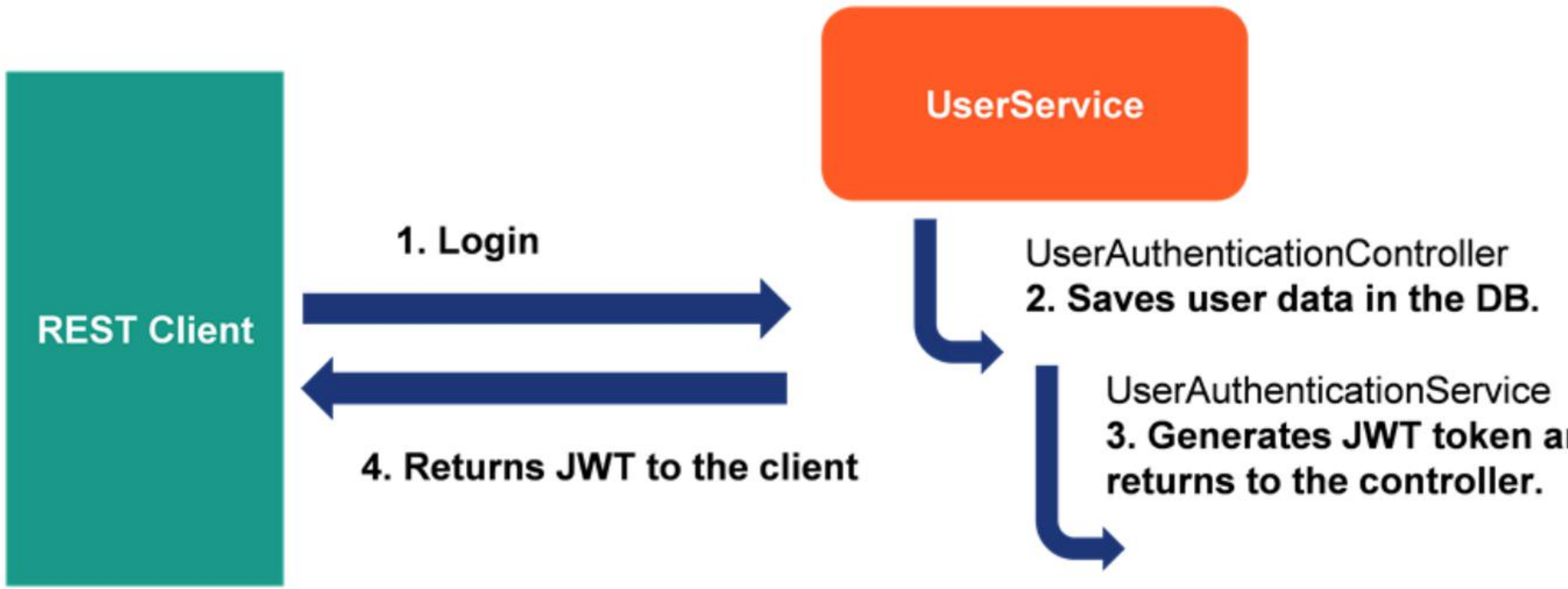
Slide Note

Data Flow of an Application Using JWT



Slide Note

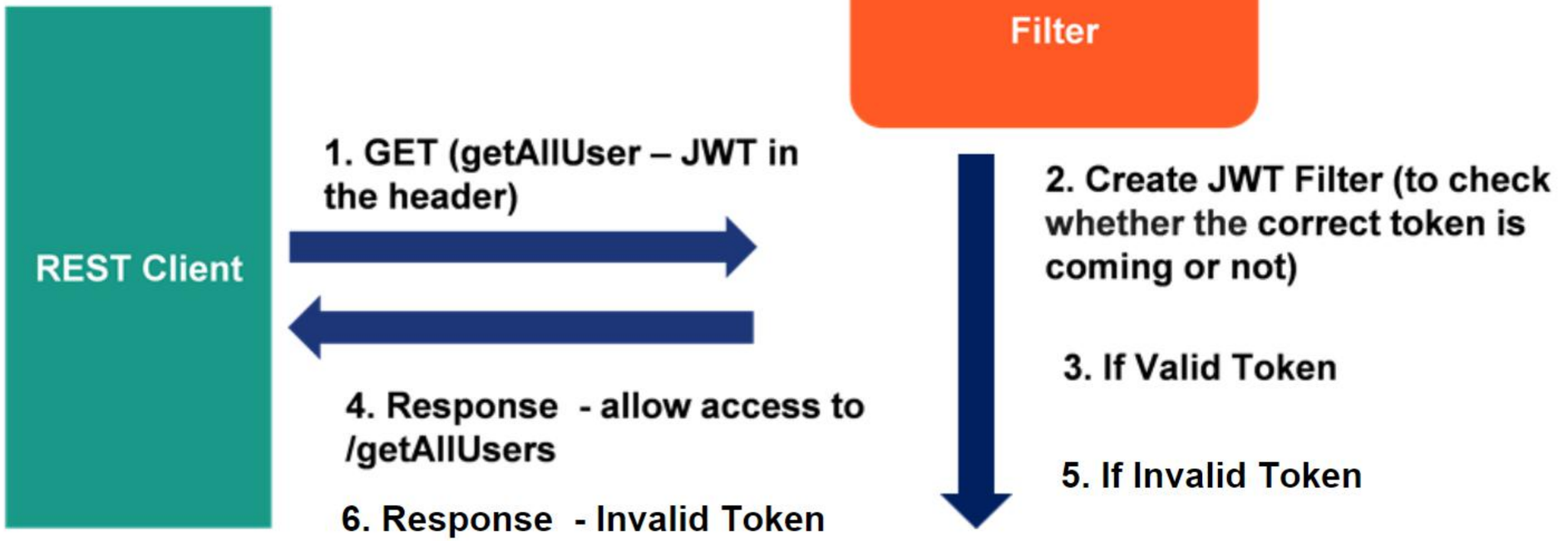
Flow Diagram – To Generate the Token



Slide Note

Access Privileged REST Endpoints

The GET mapping endpoint /`getAllUsers()` is a privileged endpoint, i.e., only requests with a JWT can access the endpoint.



Slide Note

A **filter** is an object that is invoked at the preprocessing and postprocessing of a request.

The filter used here is GenericFilterBean, which is a class.

It implements the filter interface.

Usage of Filter

• Recording all incoming requests

• Logs the IP addresses of the computers from which the requests originate

• Conversion

• Data compression

• Encryption and decryption

• Input validation, etc.

Advantage of Filter

• Filter is pluggable.

• One filter don't have dependency onto another resource.

• Less maintenance.

Verification of JWT Token by Filter

- A filter is an object that intercepts the incoming request and performs the preprocessing and postprocessing of the request.