JWT Introduction
(JSON Web Token)

By Ramesh Fadatare (Java Guides)

What is JWT

- JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.
- JWT, or JSON Web Tokens (RFC 7519), is a standard that is mostly used for securing REST APIs.
- JWT is best way to communicate securely between client and server
- **JWT follows stateless authentication mechanism**

When should you use JSON Web Tokens?

- Authorization
- Information Exchange

What is the JSON Web Token structure?

- JSON Web Tokens consist of three parts separated by dots (.), which are:
- Header

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

- Payload

```
{
    "sub": "1234567890",
    "name": "John Doe",
    "admin": true
}
```

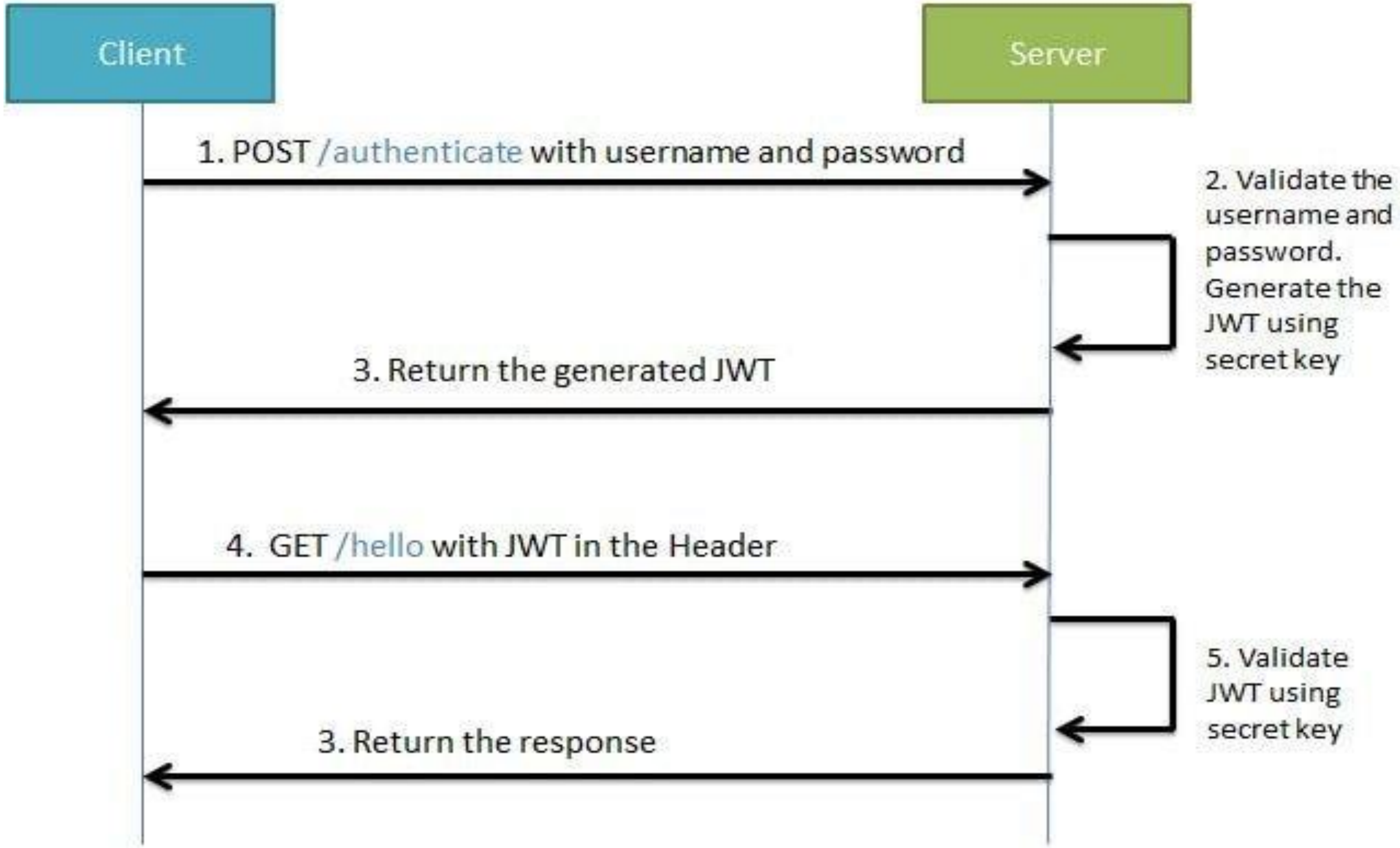
- Signature

```
HMACSHA256(
base64UrlEncode(header) + "." +
base64UrlEncode(payload),
secret)
```

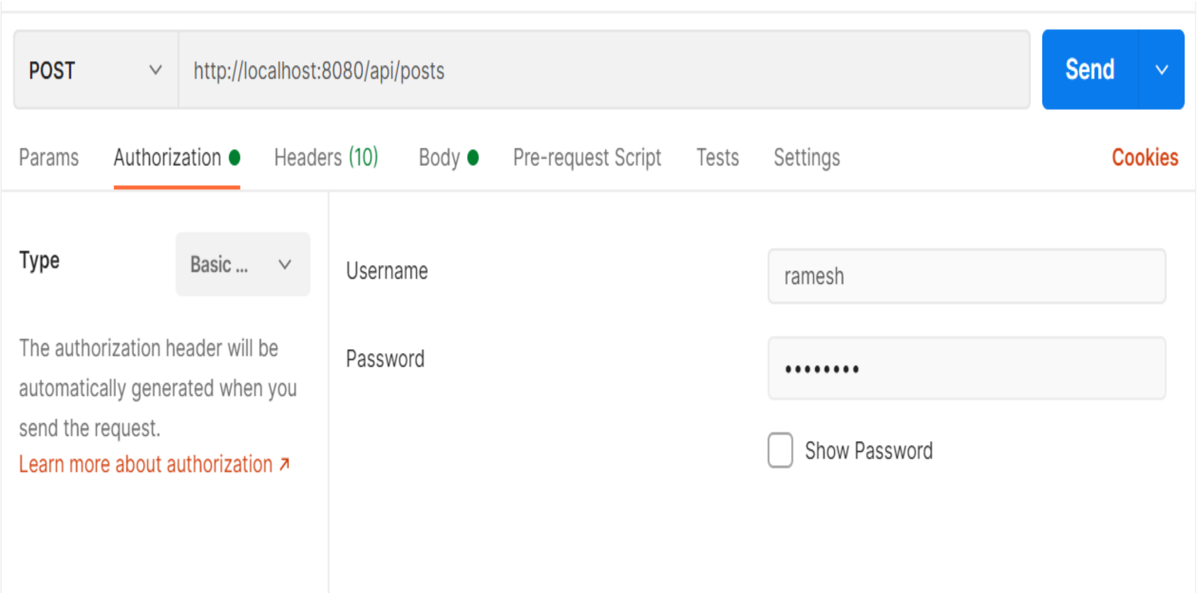
```
xxxxx.yyyyy.zzzzz
```

```
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbkBnbWFpbC5jb20iLCJpYXQiOjE2MTY1NjY5NDksImV4cCI6MTYxNzE3MTc0OX0.RVggbCFH2VGRZw9-pptLl7EKgp2BYxfOw8DXoE22MVTGJUBer600dx49UZyd-TeFvBPflOKH9Rbi8SOvzYmIAA
```

How do JSON Web Tokens work?


Client
Server
1. POST /authenticate with username and password
2. Validate the username and password. Generate the JWT using secret key
3. Return the generated JWT
4. GET /hello with JWT in the Header
5. Validate JWT using secret key
3. Return the response

Spring Security Basic Auth



POST http://localhost:8080/api/posts

Type: Basic ...

The authorization header will be automatically generated when you send the request.

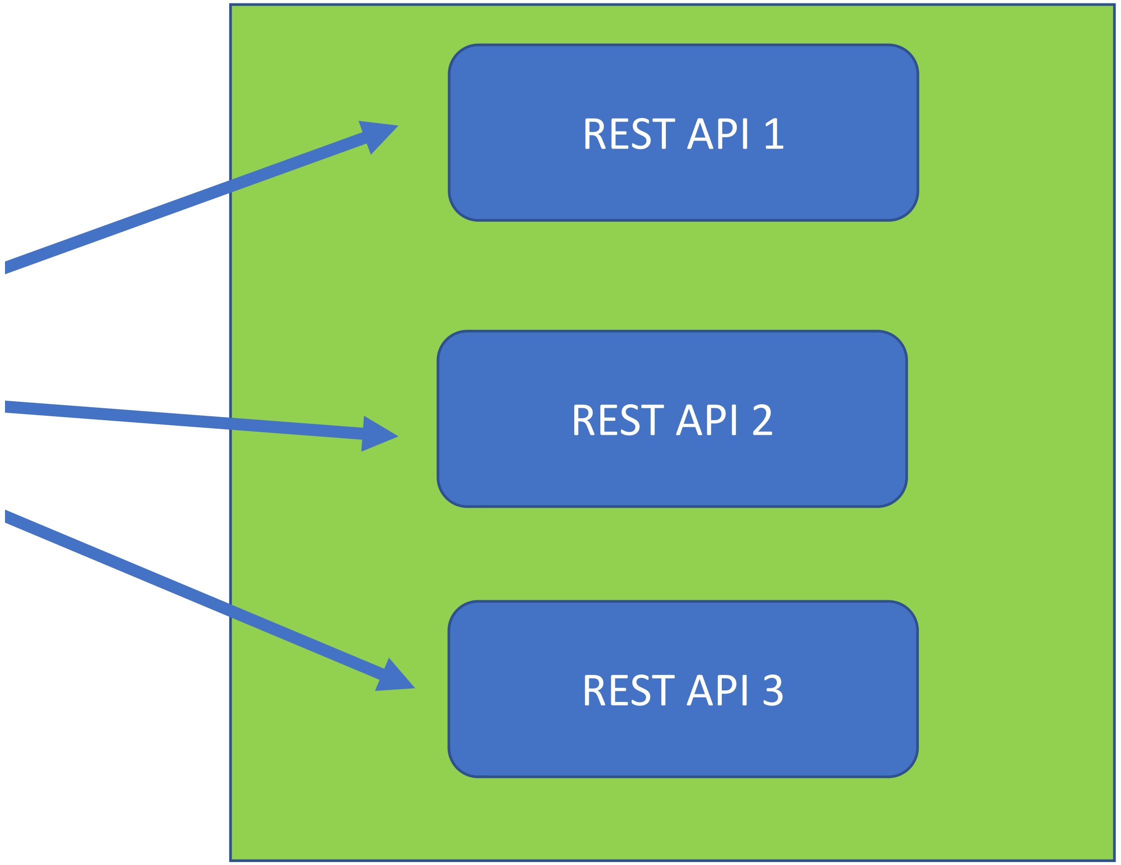
Learn more about authorization ↗

Username: ramesh

Password: ••••••••

□ Show Password

Spring Boot REST APIs



REST API 1

REST API 2

REST API 3

Development Process

Step 1: Adding JWT Dependency

Step 2: Create JwtAuthenticationEntryPoint

Step 3: Add jwt properties in application.properties file

Step 4: Create JwtTokenProvider

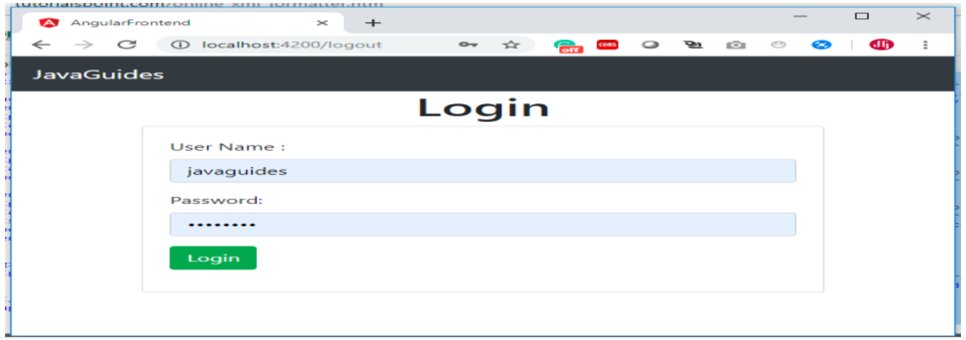
Step 5: JwtAuthenticationFilter

Step 6: Create JWTAuthResponse DTO

Step 7: Configure JWT in Spring Security Configuration

Step 8: Change login/signin API to return token to client

Spring Security + JWT



AngularFrontend

localhost:4200/logout

JavaGuides

Login

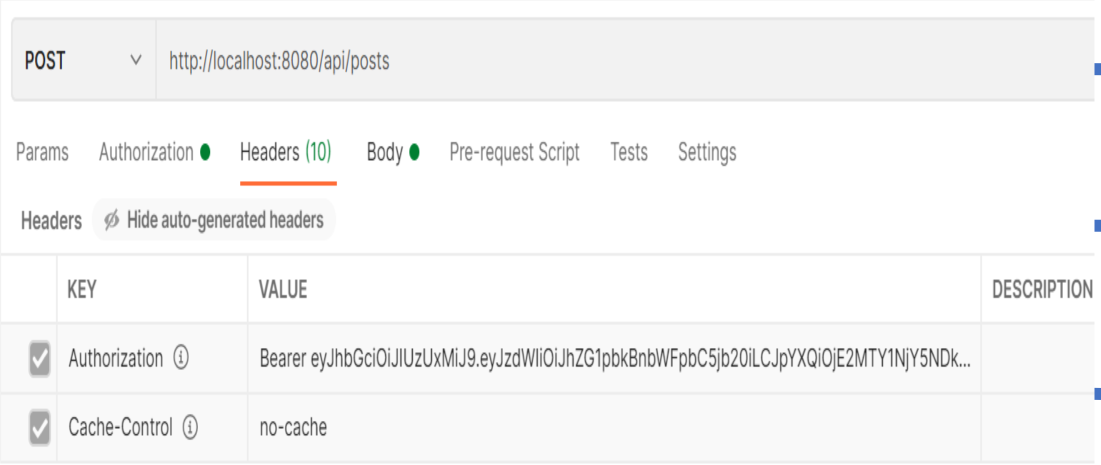
User Name :

javaguides

Password:

•••••••••

Login

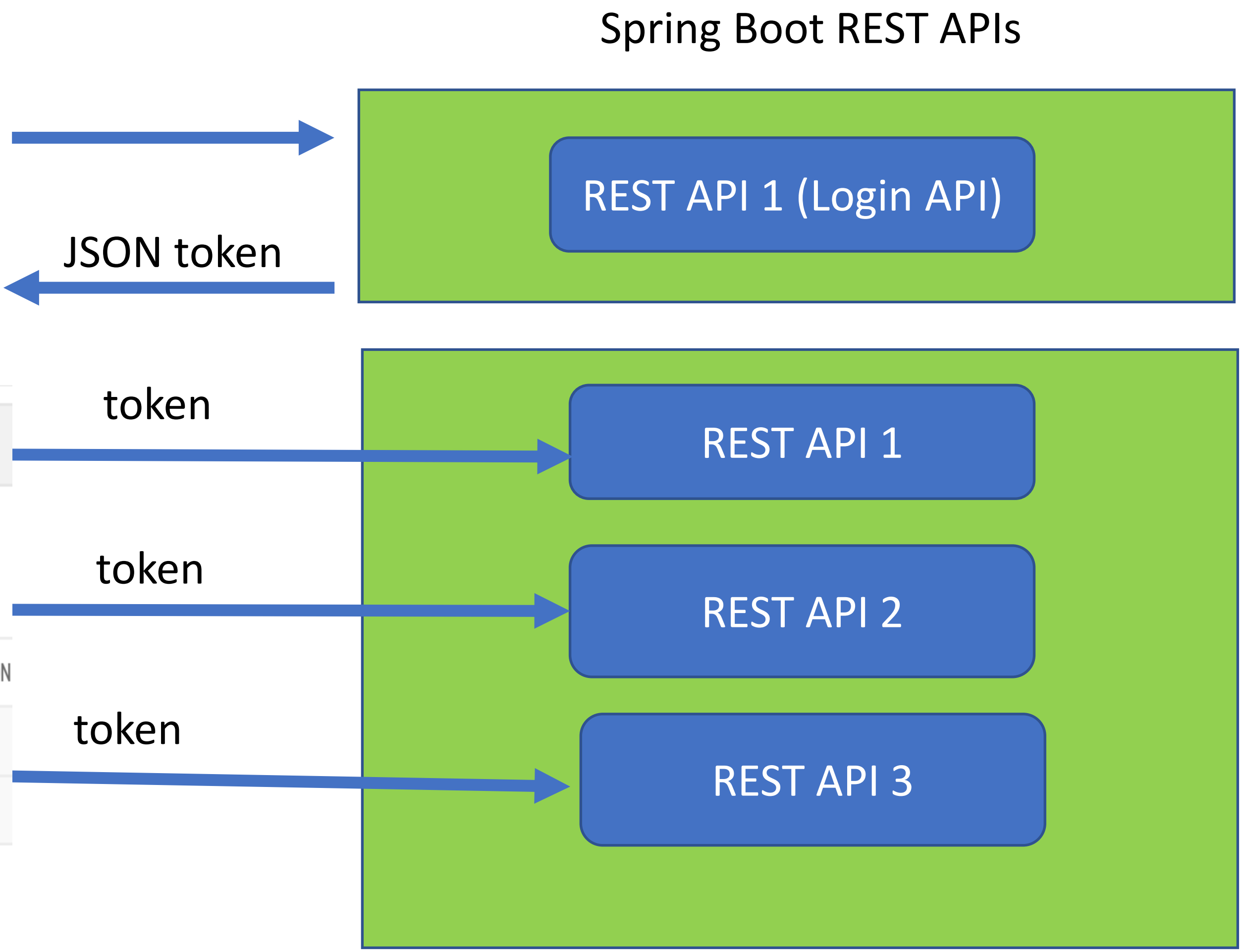


POST http://localhost:8080/api/posts

Params | Authorization ● | Headers (10) | Body ● | Pre-request Script | Tests | Settings

Headers | Hide auto-generated headers

	KEY	VALUE	DESCRIPTION
☑	Authorization ⓘ	Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbkBnbWFpbC5jb20iLCJpYXQiOjE2MTY1NjY5NDk...	
☑	Cache-Control ⓘ	no-cache	


Spring Boot REST APIs
REST API 1 (Login API)
JSON token
token
REST API 1
token
REST API 2
token
REST API 3