# Angular 10 + Spring Boot JWT Token Based Authentication Example

[[](https://medium.com/@loizen.best?source=post_page-----a7ac6c76021c--------------------------------)](https://medium.com/@loizen.best?source=post_page-----a7ac6c76021c--------------------------------)

[Loizen Best](https://medium.com/@loizen.best?source=post_page-----a7ac6c76021c--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2F687717bddba1&operation=register&redirect=https%3A%2F%2Fmedium.com%2F%40loizen.best%2Fangular-10-spring-boot-jwt-token-based-authentication-example-a7ac6c76021c&user=Loizen+Best&userId=687717bddba1&source=post_page-687717bddba1----a7ac6c76021c---------------------post_header-----------)

4 min read

·

Nov 16, 2020

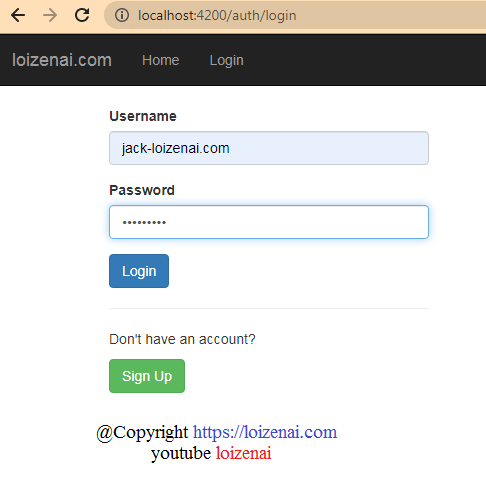
3

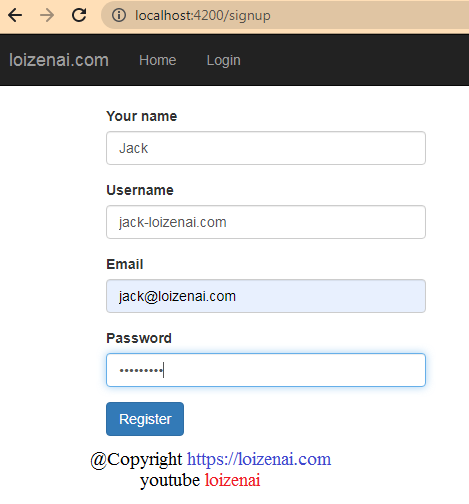
In tutorial ‘Angular 10 + Spring Boot JWT Token Based Authentication Example’, I guide you very clearly how to implement full stack example to demonistrade an jwt token based authentication flow from frontend Angular10 to backend: SpringBoot and MySQL.

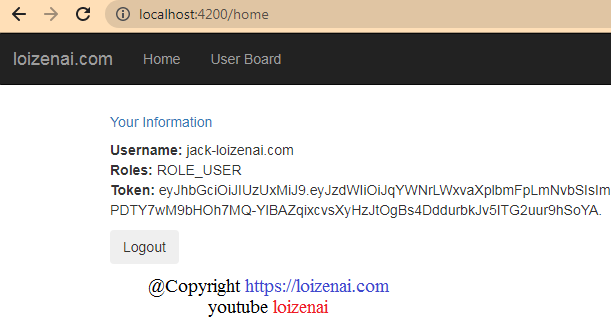


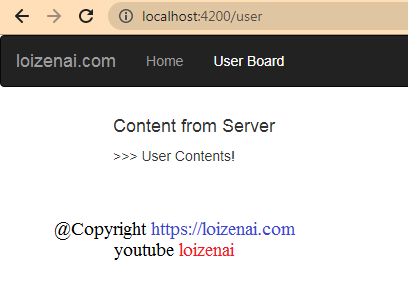
# Angular Spring Boot JWT Authentication example

We will build an application, from frontend (Angular) to backend (Spring Boot), which allows users to register, login account. This application is secured with JWT (JSON Web Token) authentication and Spring Security. Then, depending on the role of current User (user, pm or admin), this system accepts what he can access:

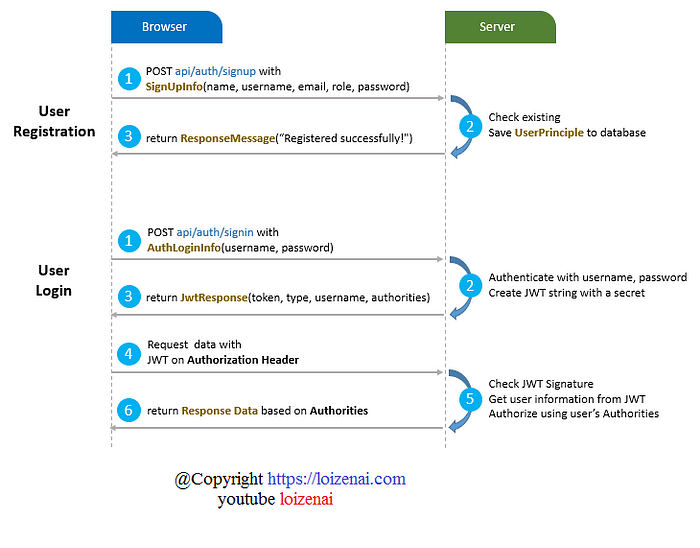






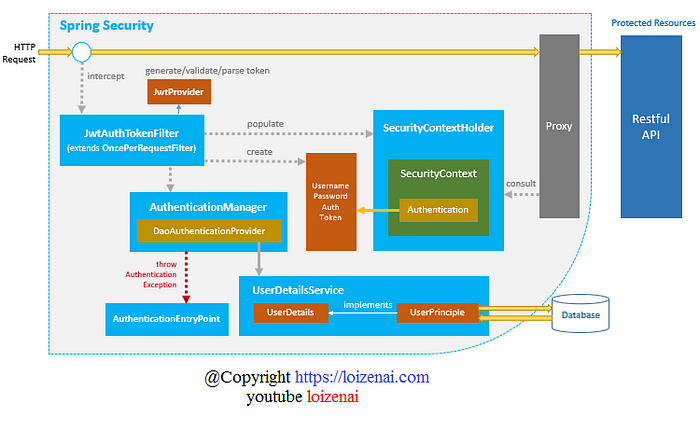


The diagram below show how our system handles **User Registration** and **User Login** processes:



# FULL STACK ARCHITECTURE — Angular 10 + Spring Boot JWT Token Based Authentication Example

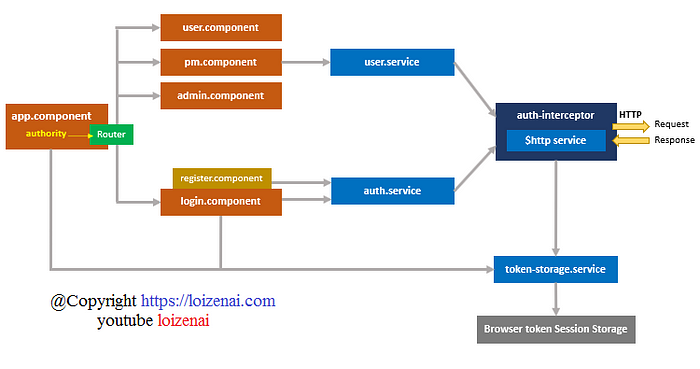
This is diagram for SpringBoot Token based authentication Security/JWT classes that are separated into 3 layers:  
– HTTP  
– Spring Security  
– REST API



– SecurityContextHolder provides access to the SecurityContext.  
– SecurityContext holds the Authentication and possibly request-specific security information.  
– Authentication represents the principal which includes GrantedAuthority that reflects the application-wide permissions granted to a principal.  
– UserDetails contains necessary information to build an Authentication object from DAOs or other source of security data.  
– UserDetailsService helps to create a UserDetails from a String-based username and is usually used by AuthenticationProvider.  
– JwtAuthTokenFilter (extends OncePerRequestFilter) pre-processes HTTP request, from Token, create Authentication and populate it to SecurityContext.  
– JwtProvider validates, parses token String or generates token String from UserDetails.  
– UsernamePasswordAuthenticationToken gets username/password from login Request and combines into an instance of Authentication interface.  
– AuthenticationManager uses DaoAuthenticationProvider (with help of UserDetailsService & PasswordEncoder) to validate instance of UsernamePasswordAuthenticationToken, then returns a fully populated Authentication instance on successful authentication.  
– SecurityContext is established by calling SecurityContextHolder.getContext().setAuthentication(…​) with returned authentication object above.  
– AuthenticationEntryPoint handles AuthenticationException.  
– Access to Restful API is protected by HTTPSecurity and authorized with Method Security Expressions.

## ANGULAR FRONT-END WITH INTERCEPTOR

In the tutorial, “Angular 10 + Spring Boot JWT Token Based Authentication Example”, we need the Angular HTTP Interceptor to add JWT Token Based for Security authentication:



– app.component is the parent component that contains routerLink and router-outlet for routing. It also has an authority variable as the condition for displaying items on navigation bar.  
– user.component, pm.component, admin.component correspond to Angular Components for User Board, PM Board, Admin Board. Each Board uses user.service to access authority data.  
– register.component contains User Registration form, submission of the form will call auth.service.  
– login.component contains User Login form, submission of the form will call auth.service and token-storage.service.

– user.service gets access to authority data from Server using Angular HttpClient ($http service).  
– auth.service handles authentication and signup actions with Server using Angular HttpClient ($http service).  
– every HTTP request by $http service will be inspected and transformed before being sent to the Server by auth-interceptor (implements HttpInterceptor).  
– auth-interceptor check and get Token from token-storage.service to add the Token to Authorization Header of the HTTP Requests.