**1. What are the steps involved in pre-processing a web request to process username header and profile loading of that that username before sending the request to the controller?**

* Extend WebSecurityConfigurerAdapter and configure http request authentication
* All the request should authenticate except the signup post request
* Configure authentication filter extending UsernamePasswordAuthenticationFilter
* Do the authentication using spring AuthenticationManager
* If authentication success generates JWT token and send to the client
* That JWT token should contain in the header in any other request coming to the system
* Then that token should validate and continue on the spring security filter chain.

**2. What are the advantages of using these filters?**

* The client sends a request to the application, and the container decides which filters and which servlet apply to it based on the path of the request URI.
* With such a filter (chain) you can basically handle every authentication or authorization problem there is in your application, without needing to change your actual application implementation.
* The expiration time can be set, because it is the best practice against secret key brute-forcing attacks.
* Can handle CSRF to improve security.
* Performance is increased because of the request doesn't travel to controller level if it is not authenticated.
* Can make use of our own security key and an algorithm.