

We use the “Thread Local Storage” pattern to store the details of who is authenticated. To indicate a user is authenticated is to set the SecurityContextHolder directly which uses a ThreadLocal to store these details. In order to get the current username, we first get a SecurityContext, which is obtained from SecurityContextHolder. This SecurityContext keeps the user details in an Authentication object, which is obtained by calling the getAuthentication() method.

When a request comes to back-end, authentication filter intercepts a user request, extracts user details from JWT token and sets it into SecurityContextHolder. This class is basically a wrapper for thread local storage, so each thread associated with a request and all child threads have access to the currently authenticated user via the static method getAuthentication(). Also with the help of this pattern, there is no need to pass user details for all methods because SecurityContextHolder provides a single global access point to the authenticated user.

