UsernamePasswordAuthenticationToken

BasicAuthenticationFilter

AuthenticationManager : ProviderManager

Setting authentication in security context

Logic for

Logic for authenticating and authorizing

AuthenticationProvider : AbstractUserDetailsAuthenticationProvider : DaoAuthenticationProvider

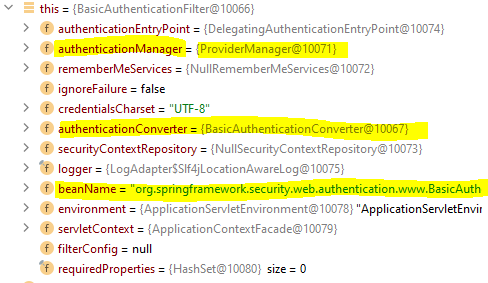
InMemoryUserDetailsManager : loadUserByUsername

**BasicAuthenticationFilter :**

1. Generates UsernamePasswordAuthenticationToken
2. Calls authenticate method of an implementation of AuthenticationProvider(in this case its ProviderManger

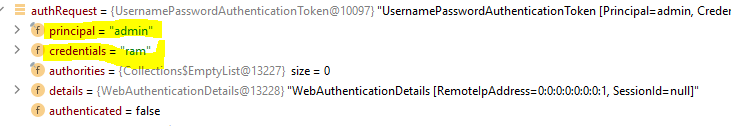
) and passes the UsernamePasswordAuthenticationToken object.

1. If authentication passes then the authentication object is stored in security context for further use
2. If authentication fails then the request is not delegated further and an exception is returned to the caller



**UsernamePasswordAuthenticationToken :**

1. Wraps the following data

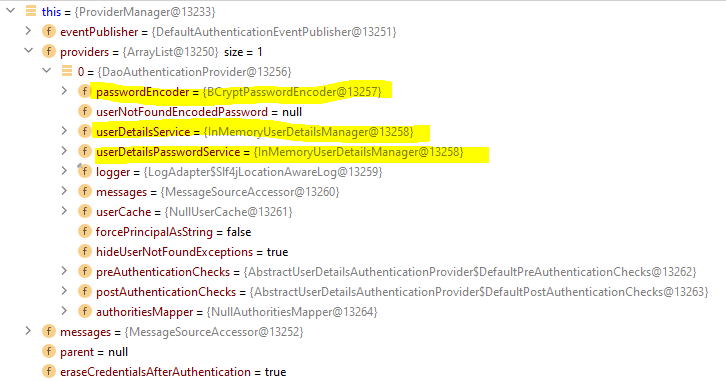


Principal is the username

Credentials is the password

**AuthenticationManager :**

1. In this case the implementation is ProviderManager
2. Contains an implementation of Authentication provider (in his case its DaoAuthenticationProvider)



**AbstractUserDetailsAuthenticationProvider :**

1. DaoAuthenticationProvider is its implementation
2. Retrieves the user details based on username with the help of DaoAuthenticationProvider
3. Authenticates the user

**DaoAuthenticationProvider :**

1. Contains PasswordEncoder, an implementation of UserDetailsService(in this case its InMemoryUserDetailsManager)
2. Calls UserService’s loadUserByUsername method to retrieve user details and create the User object

UsernamePasswordAuthenticationToken

BasicAuthenticationFilter

AuthenticationManager : ProviderManager

Setting authentication in security context

Logic for

Logic for authenticating and authorizing

AuthenticationProvider : AbstractUserDetailsAuthenticationProvider : DaoAuthenticationProvider

User-defined service class implementing UserDetailsService : loadUserByUsername

1. From the above 2 diagrams it is evident that DaoAuthenticationProvider will use InMemoryUserDetailsManager by default to load the user details from inline memory
2. If we want to write customized logic to retrieve user details then we need to write a user defined class implementing spring provided UserDetailsService