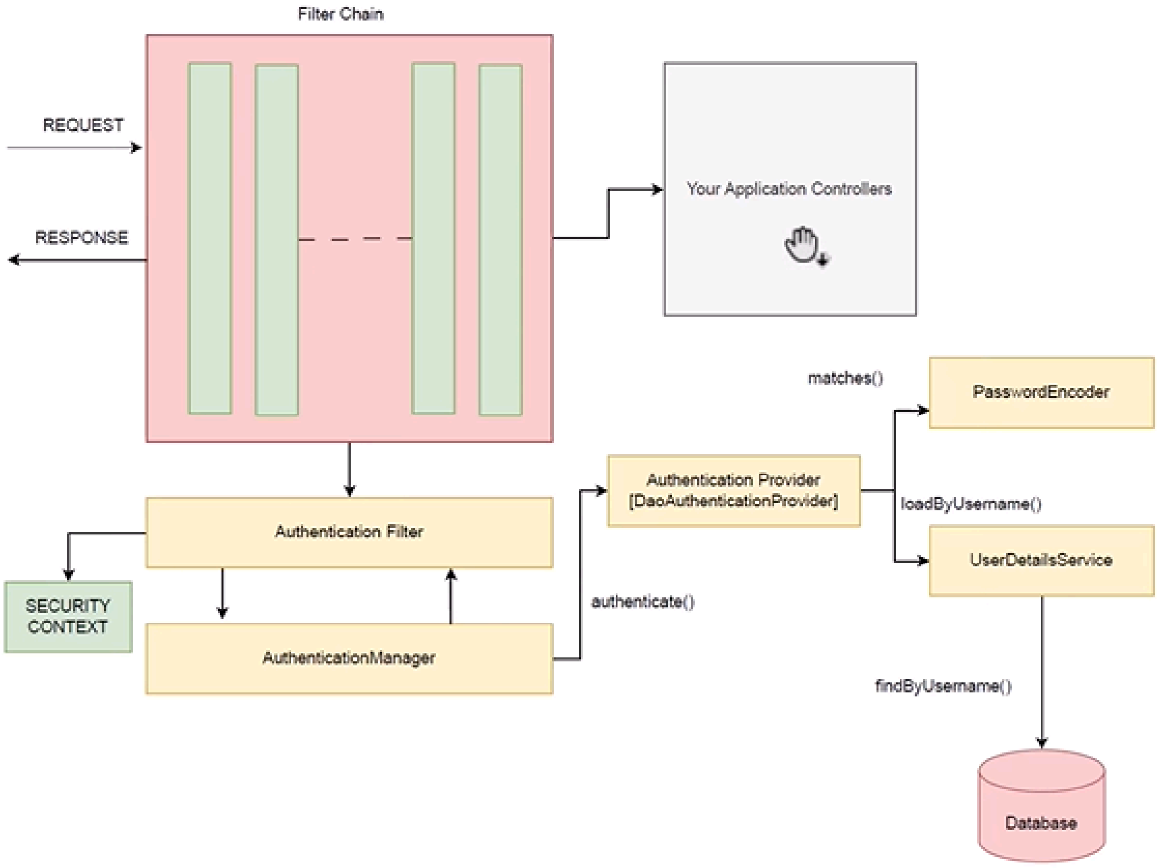
- ## How Spring Security Works ?



The diagram depicts the flow of a request through a Spring Security filter chain. Here’s a detailed explanation of each component and the flow:

1. **Request and Response:**
    - A request enters the system and eventually a response is sent back.
2. **Filter Chain:**
    - The filter chain processes the incoming request. Filters are used to perform various tasks such as authentication, logging, or input validation.
3. **Authentication Filter:**
    - One of the filters in the chain is the `Authentication Filter`. It intercepts the request to check if it contains authentication information.
4. **Security Context:**
    - The security context holds the security information for the current request, such as the authentication token. This is checked or set by the authentication filter.

5. **AuthenticationManager:**
    - The `Authentication Filter` delegates the authentication process to the `AuthenticationManager`. The `AuthenticationManager` is responsible for coordinating the authentication process by calling various `AuthenticationProviders`.
6. **Authentication Provider (DaoAuthenticationProvider):**
    - The `AuthenticationManager` uses an `Authentication Provider`, such as `DaoAuthenticationProvider`, to perform the actual authentication.
    - The `DaoAuthenticationProvider` is responsible for retrieving user details from a data source and validating the credentials.
7. **UserDetailsService:**
    - The `DaoAuthenticationProvider` calls the `UserDetailsService` to load the user-specific data.
    - The `UserDetailsService` has a method `loadByUsername()` which fetches user details from a database or another persistent storage.
8. **Database:**
    - The `UserDetailsService` interacts with the database to retrieve user details by invoking the `findByUsername()` method.
9. **PasswordEncoder:**
    - The retrieved user details include encoded passwords. The `PasswordEncoder` is used to compare the provided password with the encoded password stored in the database.
    - The `PasswordEncoder` has a method `matches()` which checks if the raw password matches the encoded password.
10. **Your Application Controllers:**
    - If the authentication is successful, the request is forwarded to the application controllers for further processing.
    - If the authentication fails, an appropriate response (such as an error message) is sent back to the client.
11. **Security Context Update:**
    - After successful authentication, the security context is updated with the authentication token, so subsequent requests can be processed without needing to re-authenticate.
12. **Response:**
    - After processing the request, a response is sent back through the filter chain.

In summary, the flow involves intercepting requests, authenticating them using a combination of filters, managers, providers, and services, and then allowing authenticated requests to reach the application controllers for further processing. Unauthenticated requests are denied or redirected appropriately.