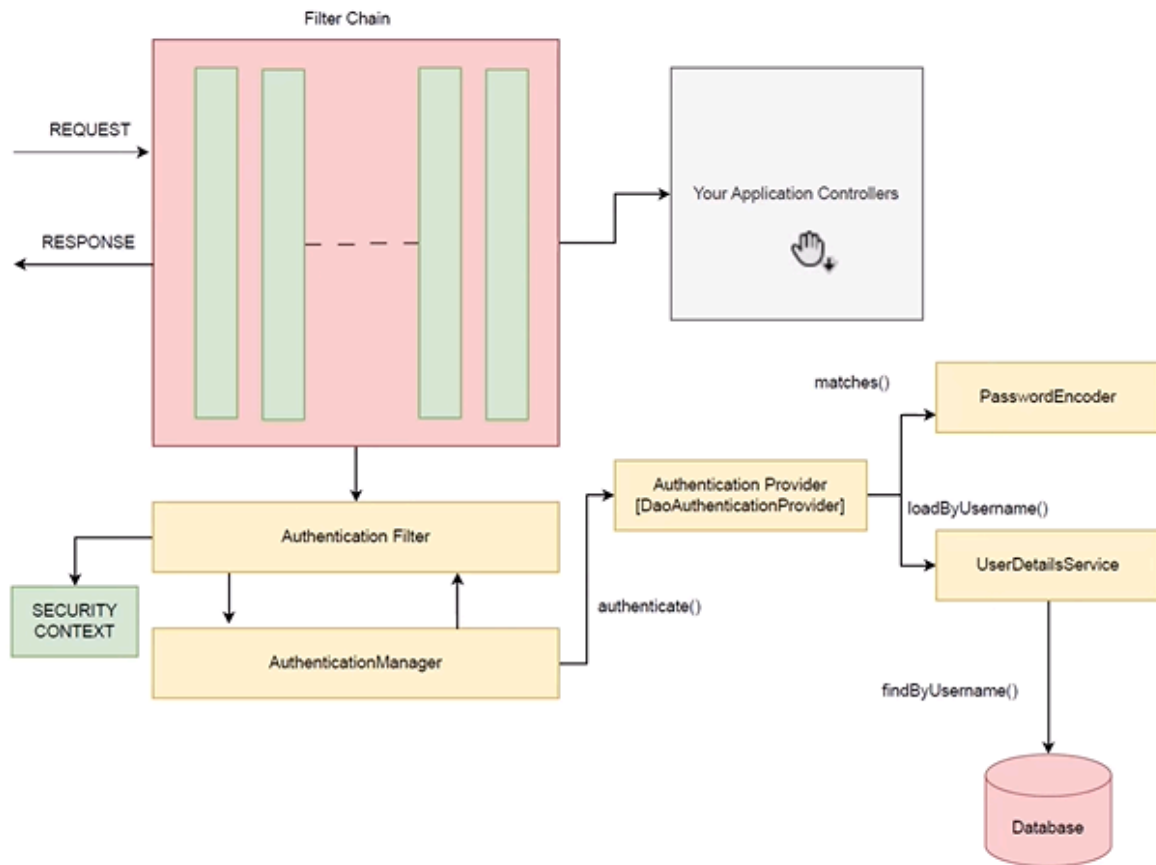


## How Spring Security Works



---

## How Spring Security Works

### 1. Request Interception:

- A request enters the application and passes through a series of filters known as the filter chain.
- Filters in this chain handle various tasks, including security-related processes.

### 2. Authentication Filter:

- If the request contains login credentials (username and password), the Authentication Filter intercepts it.
- The Authentication Filter extracts the credentials and creates an Authentication object.

3. **Authentication Manager:**
  - The Authentication object is handed over to the AuthenticationManager.
  - The AuthenticationManager delegates the task of verifying the credentials to an AuthenticationProvider.
4. **Authentication Provider:**
  - The AuthenticationProvider is responsible for validating the username and password.
  - It requires two components:
    - **PasswordEncoder:** Encodes and matches the password.
    - **UserDetailsService:** Loads user details from the database.
5. **Password Encoding:**
  - The PasswordEncoder ensures the password from the request matches the stored password by encoding and decoding it appropriately.
6. **User Details Service:**
  - The UserDetailsService retrieves user details, including roles and permissions, from the database.
7. **Authentication Validation:**
  - The AuthenticationProvider validates the credentials and updates the Authentication object with additional details (e.g., roles).
8. **Security Context:**
  - Upon successful authentication, the Authentication object is stored in the Security Context.
  - The Security Context is accessible throughout the request, defining what the authenticated user is allowed to do.
9. **Access to Controllers:**
  - With the Security Context populated, the user is granted access to application controllers based on their authenticated status and roles.

## DaoAuthenticationProvider

- The DaoAuthenticationProvider is a common AuthenticationProvider used to validate users against a database.
- Other types include:
  - InMemoryAuthenticationProvider
  - LDAPAuthenticationProvider
  - JDBCAuthenticationProvider

## **Summary Flow:**

1. User submits login credentials.
2. Request passes through the filter chain to the Authentication Filter.
3. Authentication Filter creates an Authentication object.
4. AuthenticationManager delegates authentication to AuthenticationProvider.
5. AuthenticationProvider uses PasswordEncoder and UserDetailsService to validate credentials.
6. On successful validation, the Security Context is updated.
7. User gains access to controllers based on their authenticated status.