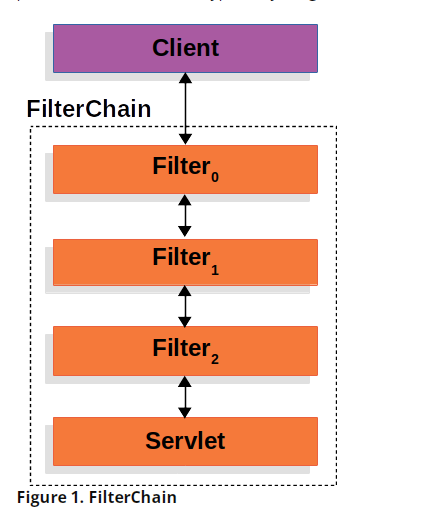
springSecurityFilterChain

Enables Spring Security’s default configuration, which creates a servlet Filter as a bean named springSecurityFilterChain

This bean is responsible for all the security (protecting the application URLs, validating submitted username and passwords, redirecting to the log in form, and so on) within your application.

Creates a **UserDetailsService** bean with a username of user and a randomly generated password that is logged to the console.

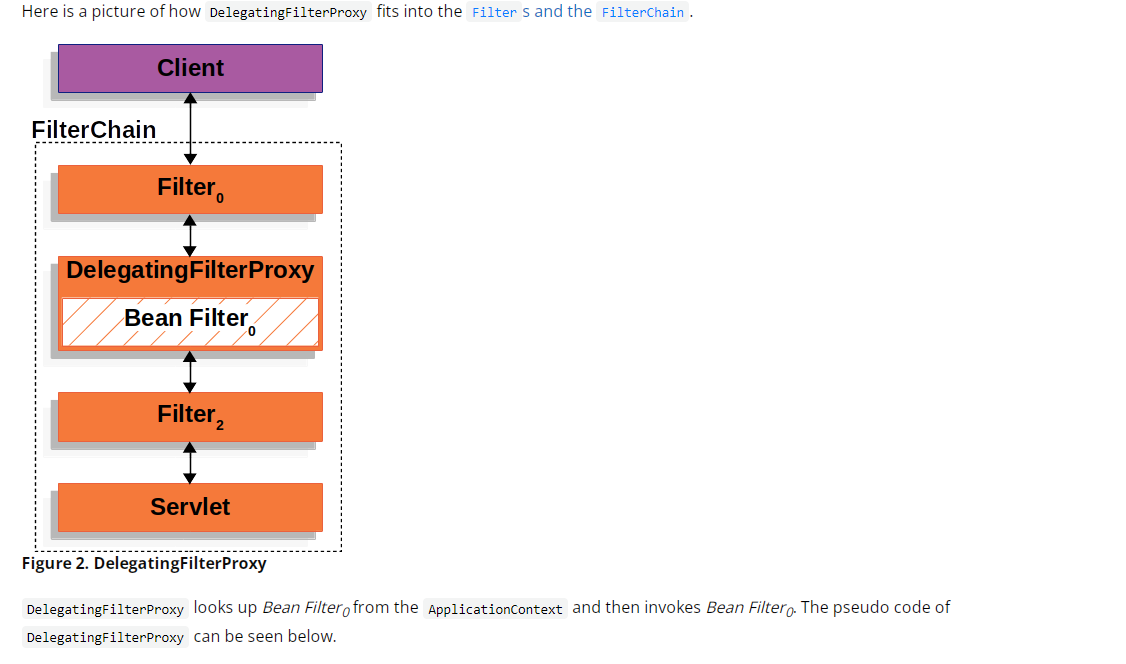
Registers the Filter with a bean named springSecurityFilterChain with the Servlet container for every request.



**9.2. DelegatingFilterProxy**

Spring provides a Filter implementation named [DelegatingFilterProxy](https://docs.spring.io/spring-framework/docs/5.3.14/javadoc-api/org/springframework/web/filter/DelegatingFilterProxy.html) that allows bridging between the Servlet container’s lifecycle and Spring’s ApplicationContext

DelegatingFilterProxy can be registered via standard Servlet container mechanisms, but delegate all the work to a Spring Bean that implements Filter.



**public** **void** **doFilter**(ServletRequest request, ServletResponse response, FilterChain chain) {

*// Lazily get Filter that was registered as a Spring Bean*

*// For the example in* [*DelegatingFilterProxy*](https://docs.spring.io/spring-security/site/docs/5.5.4/reference/html5/#servlet-delegatingfilterproxy-figure) *delegate is an instance of Bean Filter0*

Filter delegate = getFilterBean(someBeanName);

*// delegate work to the Spring Bean*

delegate.doFilter(request, response);

}