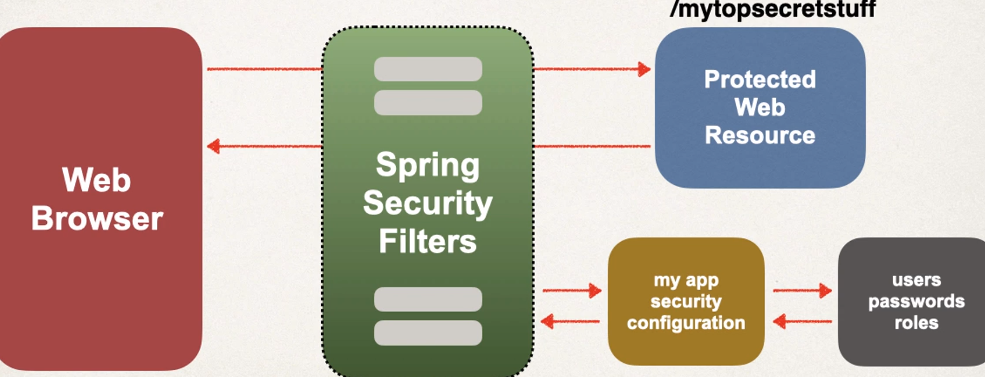
Spring Security definisce un framework per la sicurezza. È implementato utilizzando filtri Servlet in background. È possibile utilizzare la sicurezza dichiarativa o programmatica. Fondamentalmente, i filtri Servlet vengono utilizzati per pre-processare e post-processare la richiesta web. Questi filtri possono instradare la richiesta web in base alla logica di sicurezza e Spring fornisce la maggior parte delle funzionalità di sicurezza con i filtri Servlet. Diamo un'occhiata a Spring Security con alcuni diagrammi. A sinistra c'è un browser web e a destra c'è una risorsa web protetta, come le mie cose top secret. Il browser sta cercando di accedere alle vostre informazioni top secret e di ottenere i risultati. È qui che entra in gioco Spring Security.

  
Questi filtri di sicurezza a molla intercettano le richieste, le pre-elaborano e poi verificano se l'utente può effettivamente accedere alla risorsa protetta.Spring Security esaminerà la configurazione della sicurezza dell'applicazione e poi guarderà anche le password e i ruoli dell'utente presenti nel database, per vedere se l'utente è autenticato e se è autorizzato ad accedere alla risorsa web. E tutto questo avviene in , grazie a Spring Security.

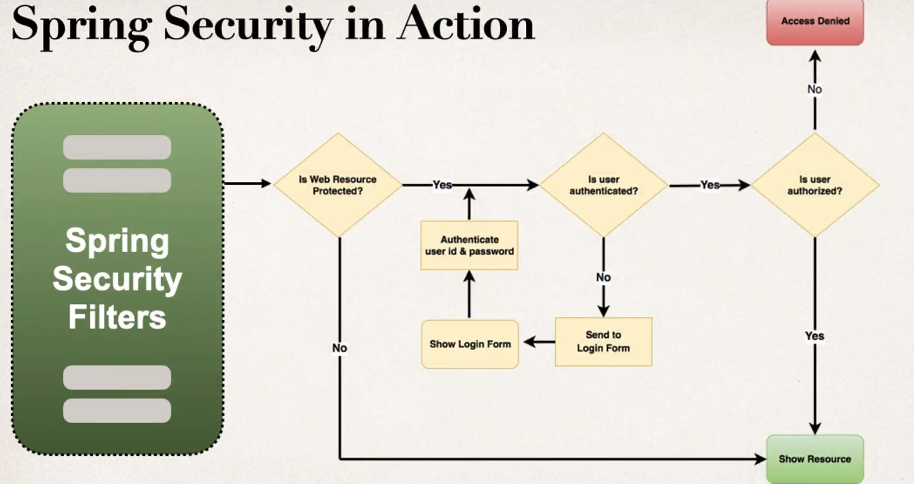
Ok, Spring Security in azione con il diagramma di flusso. Abbiamo questi filtri di Spring Security e quindi in pratica diremo: "Ehi, questa risorsa web è protetta? Sì, l'utente è autenticato? Se non sono autenticati, li invieremo al modulo di accesso, mostreremo il modulo di accesso e prenderemo l'ID utente e la password

per autenticarli. Verificheremo se l'ID utente e la password sono validi in base alle informazioni memorizzate nel nostro sistema. Poi, se superano un test, quindi sono autenticati, controlliamo: questo utente è autorizzato? Hanno l'autorizzazione ad accedere a questa risorsa? Proprio come nel vostro lavoro, avete un badge di sicurezza o all'università avete un tesserino studentesco che vi permette di entrare nell'edificio principale. Tuttavia, potrebbe non essere possibile accedere a tutte le stanze dell'edificio.

Tutto dipende dal livello di accesso o dal ruolo di autorizzazione. Solo perché si dispone di un buon ID utente e di una password, non significa che si abbia accesso a tutto. Ci sono ancora ulteriori livelli di sicurezza e Spring li supporta utilizzando i ruoli di sicurezza. Se non si ha il ruolo indicato, l'accesso è negato. Oppure, se avete il ruolo assegnato, vi mostrerà effettivamente la risorsa e vi darà accesso

a questo sito sicuro. Fondamentalmente, la sicurezza qui è in atto per tenere fuori i cattivi e permettere

ai buoni di entrare. Questo è molto bello.



Ok, **rivediamo alcuni concetti di sicurezza**. Abbiamo questa idea di **autenticazione** che è fondamentale questo gestisce la verifica dell'ID utente e della password con le credenziali memorizzate nell'applicazione o nel database, **e** poi c'è l'idea **dell'autorizzazione**. Qui si controlla se l'utente ha un ruolo autorizzato per accedere a quella risorsa.  
**2 step autenticazione, autorizzazione**

Come si può vedere, ci sono due livelli di sicurezza per il framework Spring Security.  
  
Spring Security ha dichiarato una sicurezza. Qui definiamo i vincoli di sicurezza della nostra applicazione in una configurazione**. Questo viene gestito da una configurazione interamente Java che utilizza una classe di configurazione**. E l'aspetto positivo è che fornisce la separazione delle preoccupazioni tra il codice dell'applicazione e il codice di sicurezza. E poi c'è anche la **sicurezza programmatica**. In sostanza, Spring fornisce un'API per la codifica di applicazioni personalizzate ed è qui che è possibile ottenere una maggiore personalizzazione per i requisiti specifici dell'applicazione.

Per i progetti aziendali in tempo reale, si potrebbe dire: "Ehi, Spring Security mi dà le basi. Tuttavia, nella mia azienda, abbiamo regole di business aggiuntive o logiche aggiuntive che dobbiamo implementare. Beh, non è un problema. È possibile estendere facilmente il framework per farlo e inserire

la propria implementazione di sicurezza personalizzata. Per abilitare Spring Security, la prima cosa da fare **è modificare il pom. xml e in questo file aggiungeremo la dipendenza per la sicurezza di Spring Boot Starter;** aggiungendo questa dipendenza, Spring Boot proteggerà automaticamente tutti i end-point dell'applicazione.

log offerto da spring security → { A questo punto non è necessaria alcuna codifica aggiuntiva. Tuttavia, impareremo a personalizzare la configurazione un po' più avanti. Una volta che la nostra applicazione viene eseguita e vi si accede, la nostra applicazione è protetta. Di conseguenza, Spring Security ci chiederà di effettuare il login. Ora, per impostazione predefinita, il nome utente è user e la password generata si trova nei log dell'applicazione. Questo è solo un test. In seguito, impareremo altre tecniche per personalizzare l'ID utente e la password, compresa la memorizzazione delle informazioni nel database. Al momento, però, ci limitiamo a considerare i valori predefiniti per il momento. Va bene. Ora, quando si accede , ci sono diverse opzioni per accedere all'applicazione. È possibile utilizzare l**'autenticazione di base HTTP.** C'è anche il supporto per un modulo di login predefinito. Quindi Spring vi darà un modulo di login predefinito.