In questo file trattiamo come configurare correttamente, spring security

1) Il primo passo è creare una classe java che ci farà da configurazionespring securityu e poi aggiungeremo

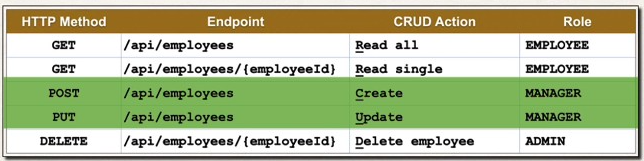
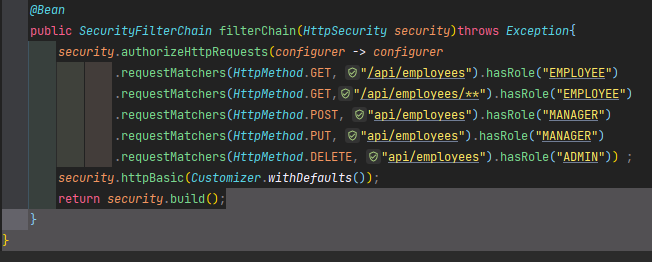
gli utenti, le password e i ruoli. La creazione di una configurazione di spring secury consiste nel marcare la

classe con la notation @Configuration, e all’interno impostermo le nostre configurazioni di sicurezza.

Useremo InMemoryUserDetailManager è una classe fornita da spring security, la quale gestisce gli utenti in memoria.   
  
Questo tipo di classe consente di memorizzare utenti e ruoli direttamente nel codice senza doverli memorizzare in un db.  
  
Questa classe definisce alcuni metodi ereditati dall’interfaccia DetailsUserManager, **come aggiungere,eliminare e cercare utenti e** gestione delle password.

Senza {noop} non fai niente, serve ad indicare a che la password non viene codificata  


Configurazione end-point per ruoli

Quindi, possiamo specificare il tipo di metodo http della richiesta, il percorso e i ruoli cosi come abbiamo un metodo per specificare che quel ruolo può accedere a tutte le aree.   
  
In questo modo ci è permesso gestire gli accessi agli end-point in modo da riservare a ruoli più importanti azioni come creare, aggiornare ed eliminare.   
  
  
  
Ricordiamo che il significato di ( \*\* ) in un percorso url come questo   
 api/employees/\*\* → questo **è una sorta di jolly,** che corrisponderà a tutti i percorsi secondari  
  
In questo modo sto definendo una configurazione per quanto riguarda le richieste http e in base al tipo di ruolo può essere eseguita una determinata operazione CRUD come possiamo vedere dall’immagine.  
  
In questo metodo stiamo sovrascrivendo i nostri filtri di sicurezza di spring security e quindi dobbiamo utilizzare la notazione Http di base e quindi **( httpBasic )** ed infine viene restituito security, fornendosci un’instanza della catena di filtri di sicurezza che sarà usata da spring una volta avviata l’applicazione.  
  
  
Inoltre spring security e in grado di proteggerci da attacchi CSRF   
  
Api stateless vengono utilizzate dalle operazioni CRUD E PATCH, ossia che non mantiene nessuno stato tra le richieste. in altre parole ogni richiesta deve contenere tutte le informazioni necessarie per elaborlarla e il server non mantiene alcuno stato della sessione del client, questo può essere disabilitato per le richieste CRUD