**Constructor injection**La prima cosa da fare è andare a definire la classe o l’interfaccia di dipendenza.  
Creeremo un controllore rest dentro il quale crearemo un costruttore nella nostra classe per le iniezioni, infine aggiungeremo un @GetMapping, per l’endpoint getDailyWorkout. → { Quindi avremo un interfaccia Coach dove ci sarà un metodo disponibile per tutti i vari coach il quale sarà sovrascritto in base al tipo di allenamento.  
L’allentore di tipo calcio sarà un componente che implementerà l’interfaccia e sovrascriverà il metodo, quindi questo allenatore, viene marcato con la notazione @Component sarà un candidato per l’inizione di dipendenza.  
  
@Component → marca la classe come un bean di spring, un bean non è nient’altro che una normale classe ma sarà gestita da spring, in questo modo rendendo disponibile la dependency injection.   
  
Ricordiamo che l’oggetto di tipo coach sarà iniettato solo dopo che la fabbrica di oggetti spring lo ha reso disponbile, quindi nel nostro costruttore del nostro controller sarà marcato con la notazione @Autowired e se si ha un solo costruttore, allora la notazione è facoltativa.  
  
 **Quindi dietro scatterà questo meccanismo:**   
  
Avremo il browser che parlerà con il nostro controller, il quale rende disponibile un metodo accessibile ad un endpoint specifico, questa disponibilità viene resa grazie alla chiamata che viene fatta dal controller al coach il quale prende il valore di ritorno e lo rende disonibile al browser rispondendolo.  
  
