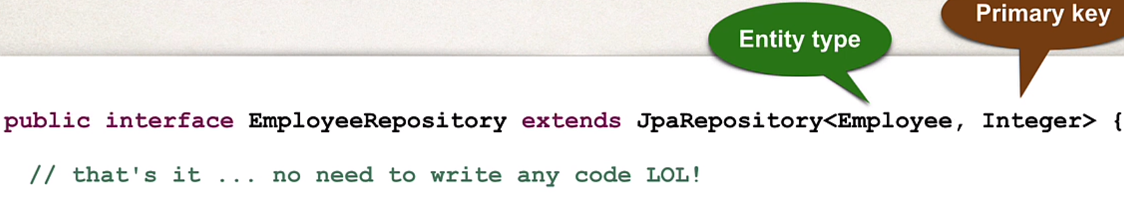
Spring data JPA  
  
Spring data JPA è una parte del framework spring che fornisce un’astrazione di alto livello per l’accesso e la gestione dei dati in un’applicazione java mediante l’uso di ORM-JPA  
  
Spring data ci consente di :

* **semplificare l’accesso ai dati** → quindi il framework genererà in automatico le query SQL necessarie per eseguire le operazioni CRUD sul database
* **Configurare facilmente le entità e i repository** → questo ci da modo di definire le operazioni CRUD le quali possono essere applicate a qualsiasi entità
* **Gestire le relazioni tra le entità** → tramite le notazioni @OneToOne, @OneToMany ecc

Spring data JPA fornisce un interfaccia chiamata **JPA Repository** la quale interfaccia espone alcuni metodi, alcuni di questi sono ereditati e questi sono i metodi CRUD  
  
   
La JPA Repository viene realizzata estendendo la classe di tipo JpaRepository<entita,chiavePrimaria>

In questo modo abbiamo accesso a tutte le operazioni CRUD.  
  


Di conseguenza non avermo più una classe DAO che farà da repository ma la nostrà entità sarà trattata anche come una repository   
  
**Inoltre** ci è concesso personalizzare le query tramite il linguaggio JPQL   
**Inoltre** possiamo rimuovere anche dai service le notazioni @Transactional in quanto JPA repository fornisce questa funzionalità di serie

Jpa fa uso degli Optional, questi sono una **caratteristica di java** che fornisce un modo più sicuro e leggibile per gestire i valori nulli, ma non sono legati specificamente a spring!

Inoltre è possibile ottenere i valori dei parametri in modo più sicuro, i parametri in una richiesta HTTP vengono trasmessi come parte dell’URL o del **corpo della richiesta** - **@bodyRequest** i quali possono essere opzionali e quindi potrebbero non essere presenti nella richiesta.  
Di conseguenza nella gestione dei parametri possiamo voler utilizzare Optional per evitare di dover gestire manualmente i valori nulli

