Repositories in Spring Data JPA

Spring Data defines a set of repository interfaces for which it generates the required code. This makes it incredibly simple to apply the repository pattern, and it also drastically reduces the amount of boilerplate code.

Activating Repositories

If you‘re using Spring Boot, Repositories are activated by default. The the package of the class that you annotated with *@SpringBootApplication* becomes the *basePackage* or you can override it using a *@EnableJpaRepositories* annotation. Spring then searches for repositories in all packages with a name that starts with the name of your *basePackage*.

If you‘re using Spring Data JPA without Spring Boot, you need to enable them by configuring the base package. You can do that by annotating your configuration class with *@EnableJpaRepositories* and providing the base package or setting the base package in your XML configuration:

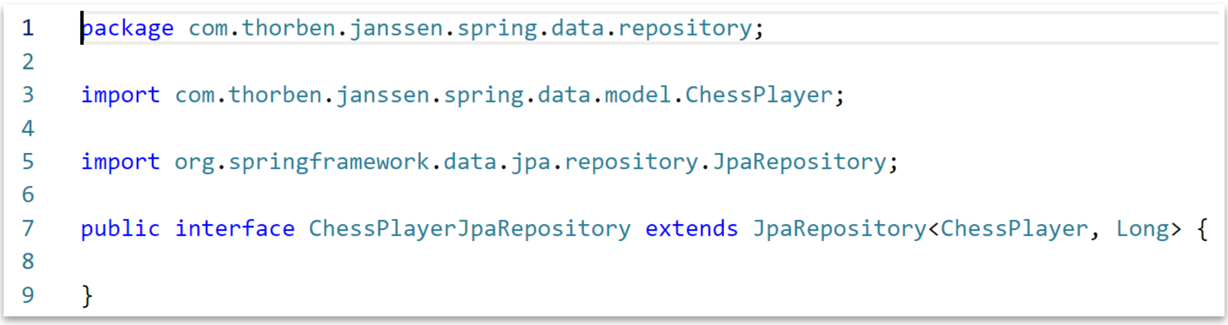
```
<beans>

    ...

    <jpa:repositories base-package="com.thorben.janssen.spring.data.repository" />
</beans>
```

Standard Repositories

To create your own repository, you need to extend one of Spring Data JPA's standard interfaces.



```java
package com.thorben.janssen.spring.data.repository;

import com.thorben.janssen.spring.data.model.ChessPlayer;

import org.springframework.data.jpa.repository.JpaRepository;

public interface ChessPlayerJpaRepository extends JpaRepository<ChessPlayer, Long> {

}
```

Spring Data JPA generates the required implementation classes for every interface that extends one of the following standard interfaces:

- *Repository*
  - A marker interface without any methods.
- *CrudRepository*
  - Extends *Repository* and adds a set of methods to create, read by id, update and delete entity objects.
- *PagingAndSortingRepository*
  - Extends *CrudRepository* and adds abstraction for pagination and sorting.
- *JpaRepository*
  - Extends *PagingAndSortingRepository* and adds JPA-specific methods persist or delete multiple entities at once, to flush the current PersistenceContext, and to use Spring Data's query by example feature.