Stored Procedures can be called using @NamedStoredProcedureQueries in spring

@Entity

@Table(name = "MYTABLE")

@NamedStoredProcedureQueries({

@NamedStoredProcedureQuery(name = "in\_only\_test",

procedureName = "test\_pkg.in\_only\_test",

parameters = {

@StoredProcedureParameter(mode = ParameterMode.IN, name = "inParam1", type = String.class)

}),

@NamedStoredProcedureQuery(name = "in\_and\_out\_test",

procedureName = "test\_pkg.in\_and\_out\_test",

parameters = {

@StoredProcedureParameter(mode = ParameterMode.IN, name = "inParam1", type = String.class),

@StoredProcedureParameter(mode = ParameterMode.OUT, name = "outParam1", type = String.class)

})

})

public class MyTable implements Serializable {

}

The key points are:

* The Stored Procedure uses the annotation @NamedStoredProcedureQuery and is bound to a JPA table.
* procedureName – This is the name of the stored procedure.
* name – This is the name of the StoredProcedure in the JPA ecosystem.
* Define the IN/OUT parameter using @StoredProcedureParameter.
* We then create the Spring Data JPA repository:
* public interface MyTableRepository extends CrudRepository<MyTable, Long> {
* @Procedure(name = "in\_only\_test")
* void inOnlyTest(@Param("inParam1") String inParam1);
* @Procedure(name = "in\_and\_out\_test")
* String inAndOutTest(@Param("inParam1") String inParam1);
* }

The key points are:

* @Procedure – the name parameter must match the name on @NamedStoredProcedureQuery
* @Param – Must match @StoredProcedureParameter name parameter
* Return types must match - so in\_only\_test is void, and in\_and\_out\_test returns String

**How to integrate struts with spring**

1.Add below code in web.xml

<filter>

<filter-name>struts2</filter-name>

<filter-class>org.apache.struts2.dispatcher.ng.filter. StrutsPrepareAndExecuteFilter</filter-class>

</filter>

<listener>

<listener-class>org.springframework.web.context. ContextLoaderListener</listener-class>

</listener>

2.Configure bean in applicationcontext.xml

In action tag of struts mention bean name instead of class name as below. helloWorldClass is the bean name in applicationcontext.xml

<struts>

<package name="default" extends="struts-default">

<action name="helloWorld" class="helloWorldClass">

<result name="SUCCESS">/success.jsp</result>

</action>

</package>

</struts>