

Spring Conversion Service Usage For Converters

In some cases we need to use Converter:

Case 1: To help simplifying bean configurations, Spring supports conversion of property values to and from text values.

Case 2: In a form, sometimes it is necessary to convert a text value to specific object, For example in a combo box there are multiple String values stored. Each value can be converted to related objects.

There are three steps to create Converter:

1. Create **Converter** class
2. Create conversion service bean in the spring config file then register the converter class
3. Refer to this bean from **conversion-service** attribute of the `<mvc:annotation-driven>`

Create Converter Class

```
1 public class AdminRoleConverter implements Converter<String,AdminRole> {
2     @Autowired
3     private AdminRoleService adminRoleService;
4     @Override
5     public AdminRole convert(String id) {
6         return adminRoleService.get(Integer.valueOf(id));
7     }
8 }
```

Conversion Service Bean

```
1 <bean id="conversionService"
2     class="org.springframework.context.support.ConversionServiceFactoryBean">
3     <property name="converters">
4         <list>
5             <bean class="com.java.spring.mvc.converter.AdminRoleConverter"/>
6         </list>
7     </property>
8 </bean>
```

Refer from `<mvc:annotation-driven>`

```
1 <mvc:annotation-driven conversion-service="conversionService"/>
```

After these three steps, you can successfully use in a jsp form:

```
1 <form:select id="slcRole" path="adminRole">
2     <form:option label="SELECT" value="{null}"/>
```

```
3     <form:options items="${roles}" itemValue="adminRoleId" itemLabel="role" />
4 </form:select>
```

Note: `adminRole` path value type is `AdminRole` class, option items consists of multiple `AdminRole` class, but displayed values are text values. Because of this, we need to convert text values to `AdminRole` objects