1. Converter 将一种类型转换成另外一种类型。

创建Converter必须实现Converter接口

public interface Converter<S, T> 从S类型转换成T类型

实现T convert(S source)方法

示例: 使请求参数里的字符串的日期对象都转换成Date对象

public class StringToDateConverter implements Converter<String, Date> {  
  
 private String datePattern;  
  
 public StringToDateConverter(String datePattern) {  
 this.datePattern = datePattern;  
 }  
  
 @Override  
 public Date convert(String s) {  
 try {  
 SimpleDateFormat dateFormat = new SimpleDateFormat(datePattern);  
 dateFormat.setLenient(false);  
 return dateFormat.parse(s);  
 } catch (ParseException e) {  
 throw new IllegalArgumentException(e);  
 }  
 }  
}

spring mvc配置文件里的配置：

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

和

<bean id="conversionService" class="org.springframework.context.support.ConversionServiceFactoryBean">  
 <property name="converters">  
 <set>  
 <bean class="com.converter.StringToDateConverter">  
 <constructor-arg type="java.lang.String" value="MM-dd-yyyy" />  
 </bean>  
 </set>  
 </property>  
</bean>

1. Formatter 把String转换成另外一种类型（Formatter更适合Spring MVC应用）

要实现 public interface Formatter<T>

实现T parse(String text, Locale locale) //利用指定的Locale将一个String解析成目标类型

String print(T object, Locate locate) //返回目标对象的字符串表示

示例：使请求参数里的字符串的日期对象都转换成Date对象

Spring mvc配置文件里的配置：

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

<bean id="conversionService" class="org.springframework.format.support.FormattingConversionServiceFactoryBean">  
 <property name="formatters">  
 <set>  
 <bean class="com.converter.DateFormatter">  
 <constructor-arg type="java.lang.String" value="MM-dd-yyyy" />  
 </bean>  
 </set>  
 </property>  
</bean>

Formatter实现类

public class DateFormatter implements Formatter<Date> {  
 private String datePattern;  
 private SimpleDateFormat dateFormat;  
  
 public DateFormatter(String datePattern) {  
 this.datePattern = datePattern;  
 dateFormat = new SimpleDateFormat(datePattern);  
 dateFormat.setLenient(false);  
 }  
  
 @Override  
 public Date parse(String s, Locale locale) throws ParseException {  
 return dateFormat.parse(s);  
 }  
  
 @Override  
 public String print(Date date, Locale locale) {  
 return dateFormat.format(date);  
 }  
}

1. FormatterRegistrar：注册多个Formatter，不再需要一个个在spirng mvc配置文件里配置。

示例：

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

<bean id="conversionService" class="org.springframework.format.support.FormattingConversionServiceFactoryBean">  
 <property name="formatterRegistrars">  
 <set>  
 <bean class="com.converter.MyFormatterRegistrar">  
 <constructor-arg type="java.lang.String" value="MM-dd-yyyy" />  
 </bean>  
 </set>  
 </property>  
</bean>

public class MyFormatterRegistrar implements FormatterRegistrar {  
  
 private String datePattern;  
 public MyFormatterRegistrar(String datePattern) {  
 this.datePattern = datePattern;  
 }  
  
 @Override  
 public void registerFormatters(FormatterRegistry formatterRegistry) {  
 formatterRegistry.addFormatter(new DateFormatter(datePattern));  
 //add more formatter  
 }  
}