**类型转换器的基本概念：**

1）收集用户的请求参数时，将请求参数传递给控制器组件。此时存在一个问题，即所 有的请求参数都是字符串类型的，但Java是强类型的语言，所以Spring MVC需要将 这些字符串请求参数转换为相应的数据类型。为此，Spring MVC提供了类型转换器， 用于完成字符串和各种数据类型之间的转换。

2）对于Spring MVC框架而言，将请求参数转换成值对象类中各个属性对应的数据类型， 这就是类型转换的意义。

**类型转换器的底层实现原理：**

1）在Spring MVC中，类型转换器底层的实现原理依靠的是Converter<S,T>接口。

2）Converter<S,T>是一个可以将数据类型转换成另一种数据类型的接口，这里的S表示 源类型，T表示目标类型。

**内置的类型转换器：**

在Spring MVC中，对于常见的数据类型，开发者无需创建自己的类型转化器，因为在 Spring MVC中已经提供了类型转换器用于完成常用的类型转换。Spring MVC提供的内置 类型转换有以下几种类型：

1. 标量转换器

·StringToBooleanConverter：String转换为boolean类型

·ObjectToStringConverter：Object转换为String类型（调用toString方法完成）

·StringToNuberConverterFactory：String转换为数字类型（如Interger、Long等）

·NumberToNumberConverterFactory：数字基本类型转换为数字包装类类型

·StringToCharacterConverter：String转换为Character，取字符串中的第一个字符

·NumberToCharacterConverter：数字基本类型转换为Character类型

·CharacterToNumberFactory：Character类型转换为数字基本类型

·StringToEnumConverterFactory：String转换为枚举类型，通过Enum.valueOf将字 符串转换为需要的枚举类型

·EnumToStringConverter：枚举类型转换为String类型，返回枚举对象的name值

·StringToLocationConverter：String类型转换为java.util.Locale类型。

·PropertiesToStringConverter：java.util.Properties类型转换为String类型，默认通 过ISO-8859-1解码

·StringToPropertiesConverter：String类型转换为java.util.Properties类型，默认使 用ISO-8859-1编码

1. 集合、数组相关转换器：

·ArrayToCollectionConverter：任意类型数组转换为任意集合（List、Set）

·CollectionToArrayConverter：任意集合转换为任意数组

·ArrayToArrayConverter：任意数组转换为任意数组

·CollectionToCollectionConverter：任意集合转换为任意集合

·MapToMapConverter：任意Map集合转换为Map集合

·ArrayToStringConverter：任意数组转换为String类型

·StringToArrayConverter：String类型转换为数组，默认通过“，”分割，且去 除字符串两边的空格（trim）

·ArrayToObjectConverter：任意数组转换为Object类型，如果目标类型和源类型兼 容，直接返回源对象；否则返回数组第一个元素并进行 类型转换。

·ArrayToObjectConverter：Object类型转换为单元素数组

·CollectionToStringConverter：任意集合（List、Set）转换为String类型

·StringToCollectionConverter：String类型转换为集合（List、Set），默认通过“，” 分割，且去除字符串两边的空格（trim）

·CollectionToObjectConverter：任意集合转换为任意Object，如果目标类型和源类 型兼容，直接返回源对象；否则返回集合的第一个 元素并进行类型转换

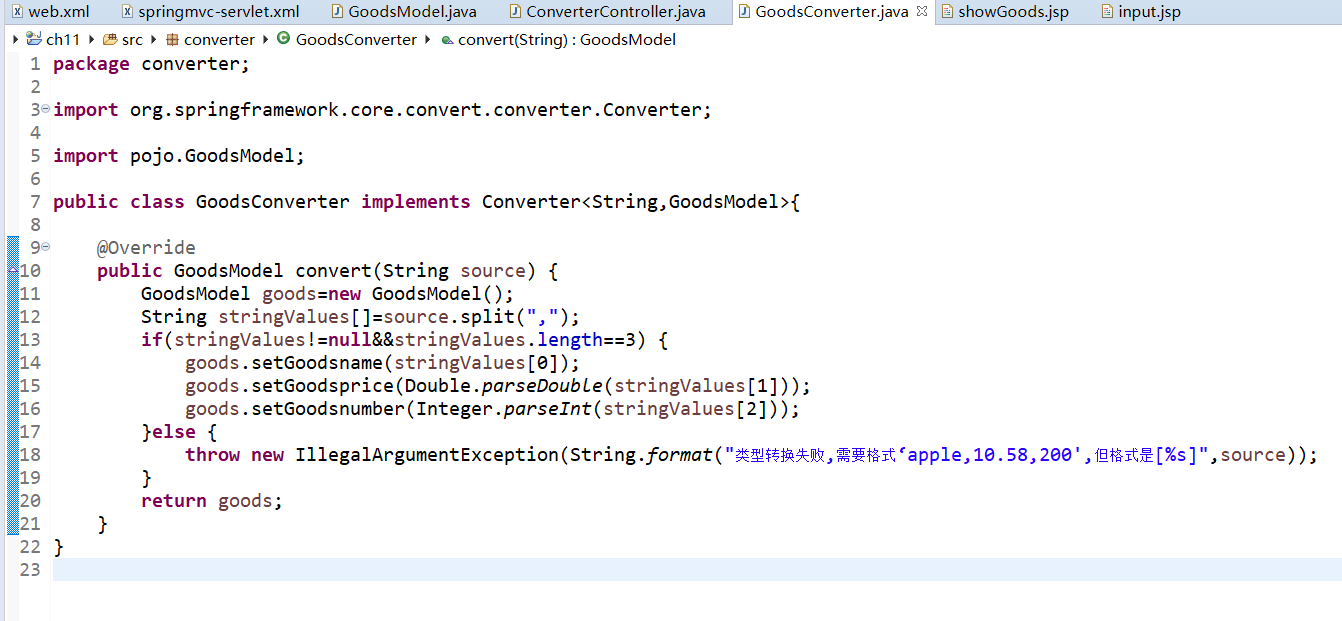
·ObjectToCollectionConverter：Object类型转换为单元素集合的类型转换

**自定义类型转换器：**

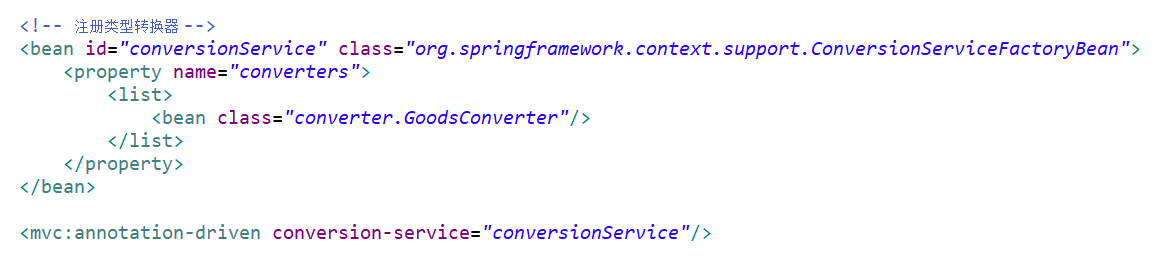
当Spring MVC内置的类型转换器不能满足需求时，开发者可以开发自己的类型转换器。

大致步骤如下：

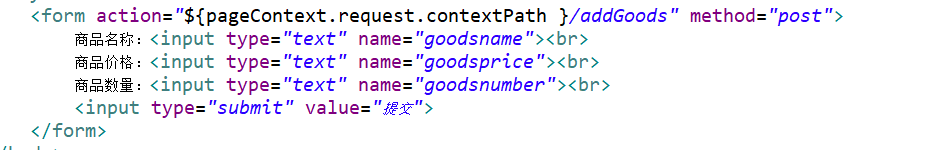
1. 自定义类型转换器类并实现Converter<S,T>接口，覆写convert(S)接口方法，convert(S) 方法的功能是将元数据类型S转换为目标数据类型T

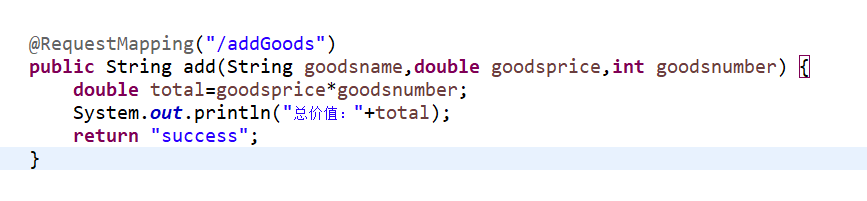


1. 定义好类型转换器后，还需要在Spring MVC的全局配置文件中对类型转换器进行注 册。



**使用类型转换器：**

使用类型转换器是自动的，在控制层接收参数时会根据处理请求方法的形参类型自动寻 找相应类型转换器进行转换。如下：



**格式化转换器：**

1. Spring MVC的Formatter<T>与Converter<S,T>一样，也是一个可以将一种数据类型转 化为另一中数据类型的接口。
2. Formatter<T>的原数据类型必须是String类型，返回一个T类型的数据。T可以是任 意数据类型，表示的是字符串转换的目标数据类型。

**Spring MVC内置的格式转换器：**

在Spring MVC中提供了几个内置的格式转换器，如下：

·NumberFormatter：实现Number与String之间的解析与格式化

·NumberFormatter：实现Number与String之间的解析与格式化（带货币符号）

·PercentFormatter：实现Number与String之间的解析与格式化（带百分号）

·DateFormatter：实现Date与String之间的解析与格式化

**自定义格式化转换器：**  
 当Spring MVC提供的内置的格式化转换器不能满足需求时，开发者可以自定义格 式 化转换器。

1. 编写一个类并实现org.springframework.format.Formatter接口，覆写Formatter接口 中parse( )方法和print( )方法

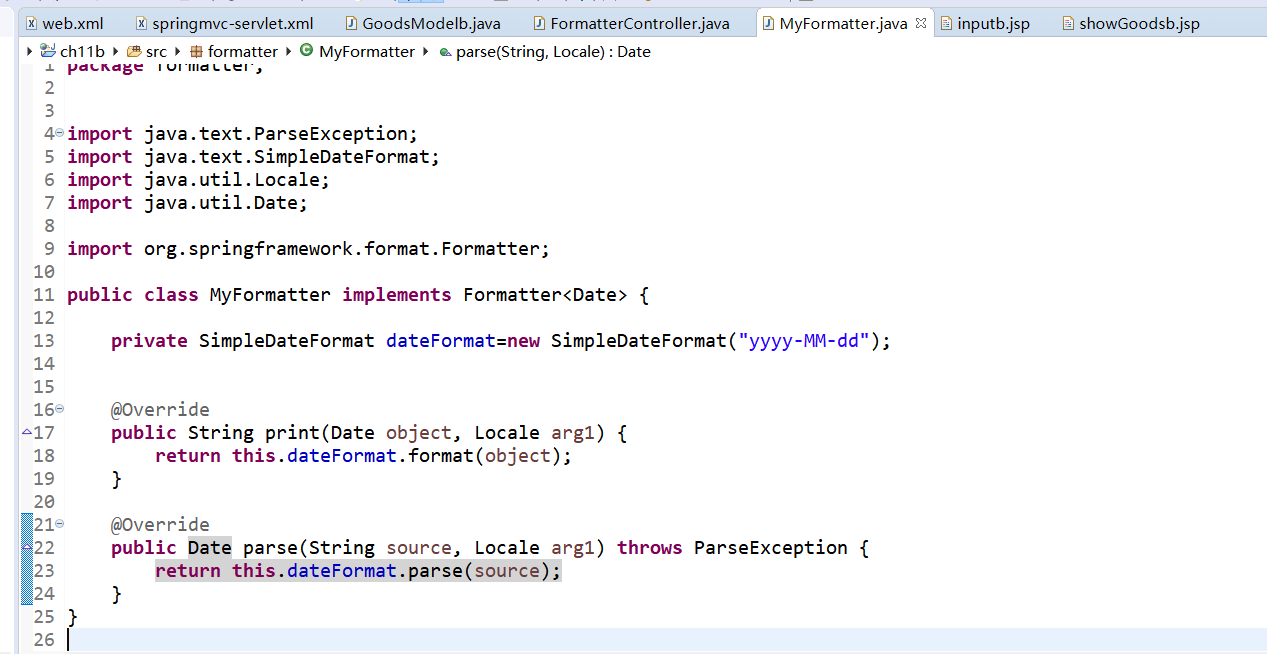
public interface Formatter<T>( )

public T parse( String s, java.util.Locale locale )

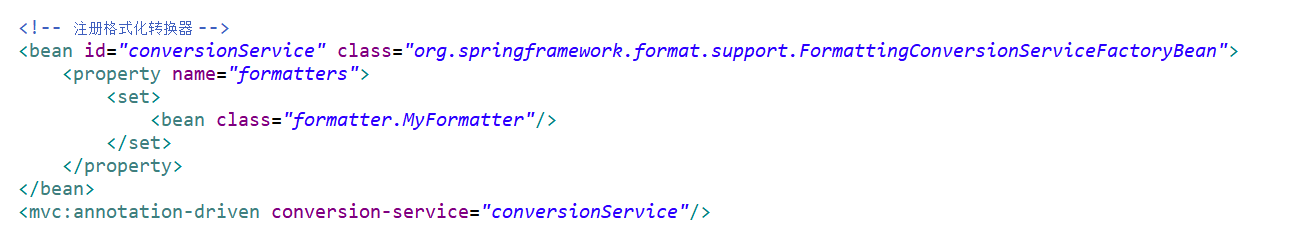
public String print( T object , java.util.Locale locale )

parse( )方法的功能是利用指定的Localse将一个String类型转换为目标数据类型；

print( )方法与之相反，用于返回目标数据类型的字符串表示。

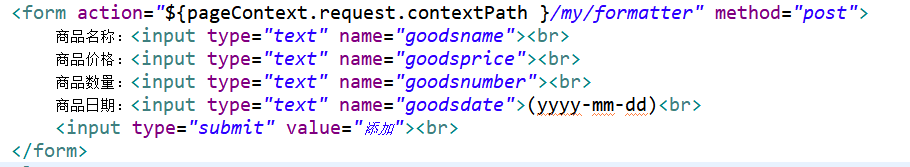


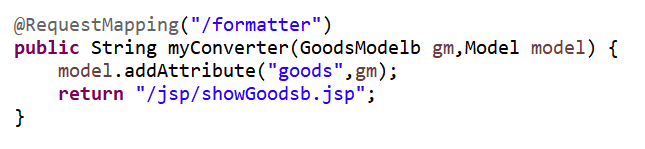
1. 在Spring MVC配置文件中注册类型转换器

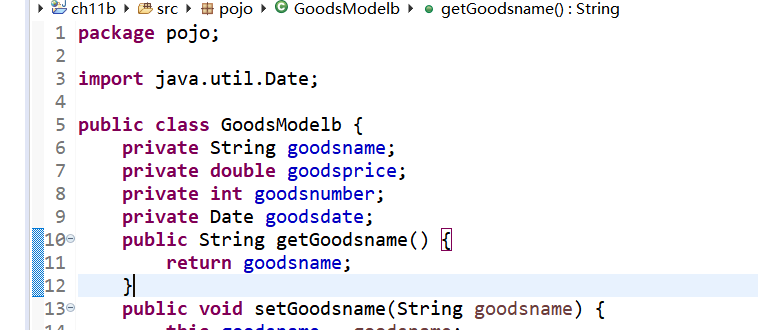


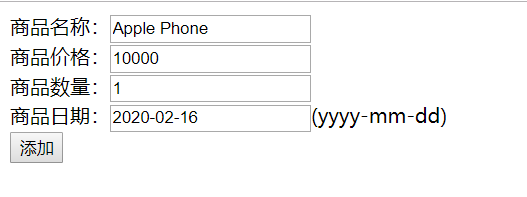
**格式化转换器的使用方法：**

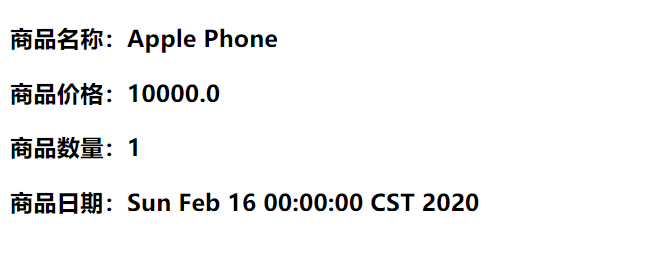
格式化转换器在请求处理方法接收形参时自动使用，如下：











可以看出，通过格式化转换器将yyyy-mm-dd字符串类型的日期转换为Date类型的日期。