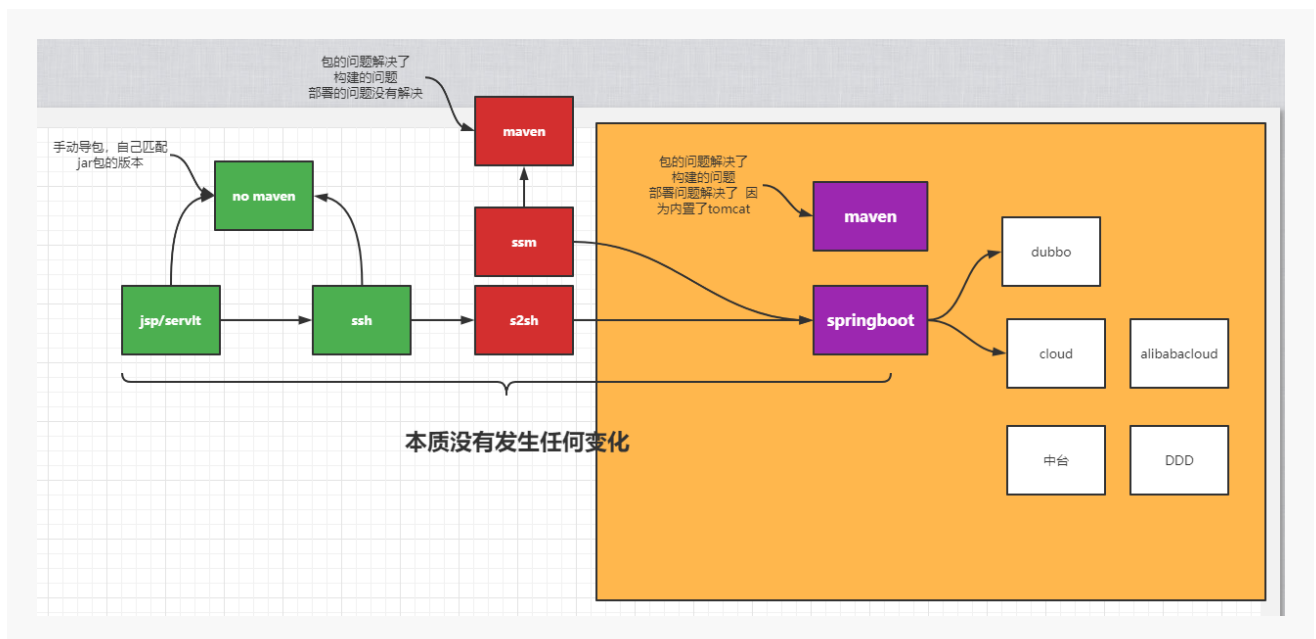
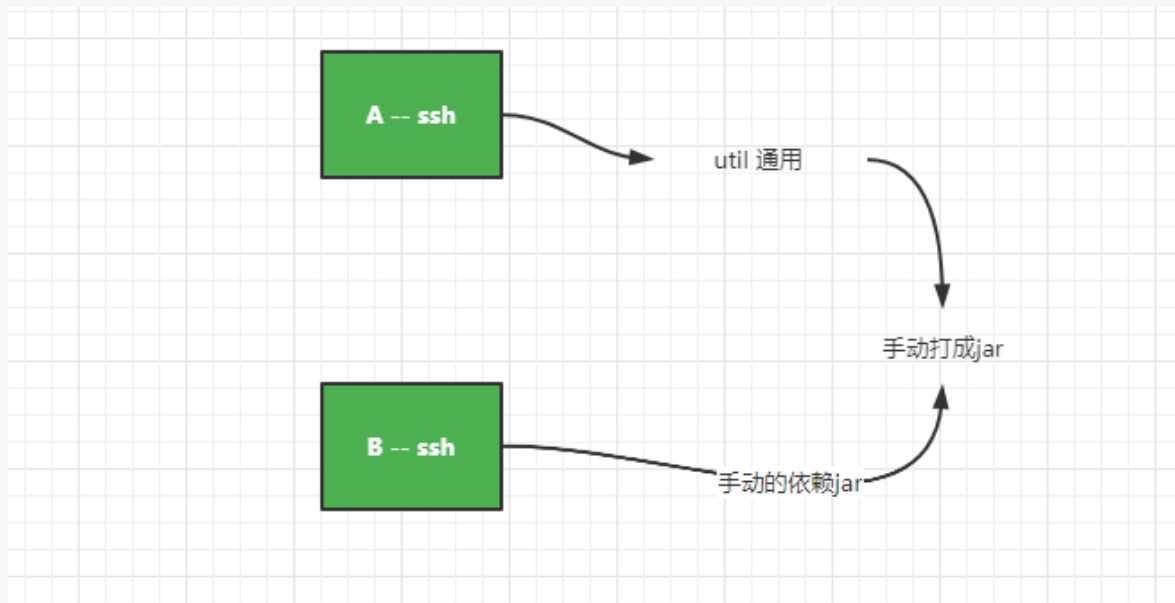


01、使用Idea搭建ssm项目过程-手动方式

分析



- 传统的开发方式必须全部由开发人员去下载jar包, 工程很难复用。(jsp/servlet / ssh)



- 自己手动去增加jar包
- 包管理和版本都需要自己去匹配
- 打包和部署的依赖非常麻烦。
- 配置文件一堆，都是问题
- 后续出现ant 和 maven 把项目构建的问题和打包解决
 - maven作用：构建 和 依赖管理
- ssm 架构
 - 构建问题使用maven解决
 - 包依赖的问题也解决了通过maven，但是版本的关系没解决
 - 依赖外部tomcat jetty容器
 - 大量配置xml配置，影响程序的开发。程序员不去关注业务，每天都在为这个烦人的配置绞尽脑汁。
- springboot开发
 - 构建问题使用maven解决
 - 包依赖的问题也解决了通过maven，通过starter机制解决版本依赖问题
 - 内置了tomcat不需要额外的配置容器
 - 零配置，用starter机制 + @Configuration、@Bean的方式解决xml的问题

02、工具的历程

桌面应用程序：c/s client-server 、 b/s

Java桌面应用程序：swing - jbuilder

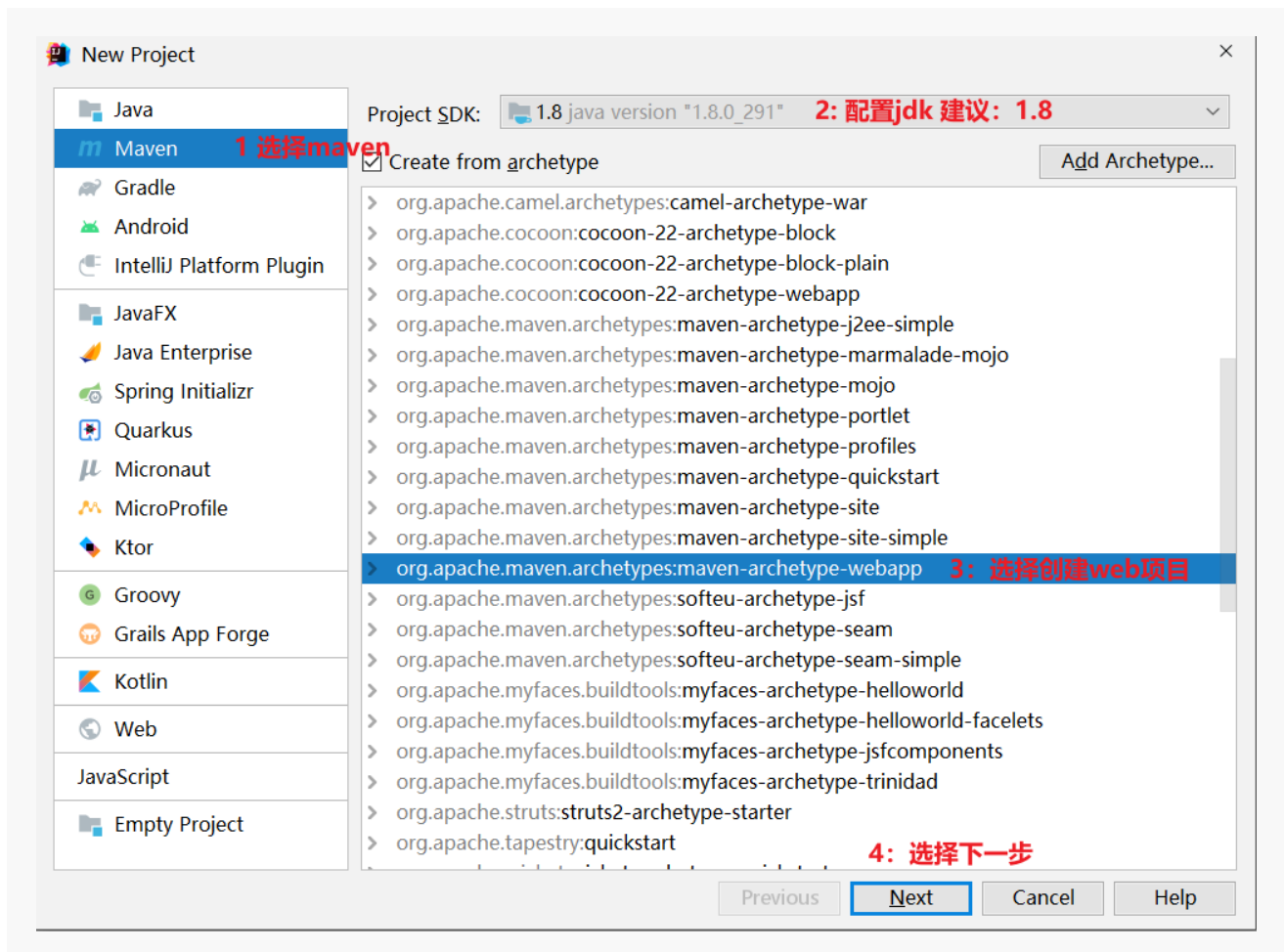
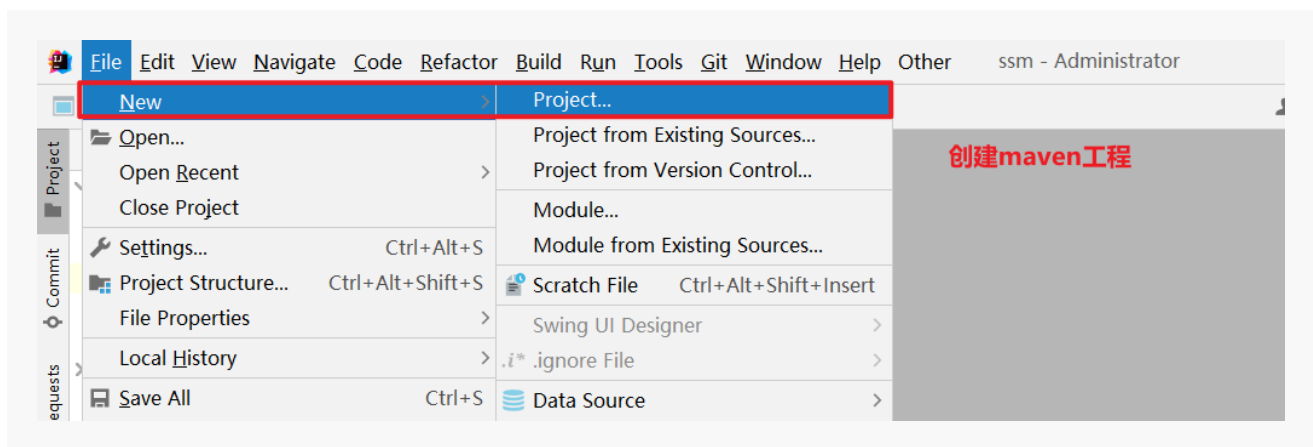
- jcreator
- jbuilder - swing / ejb
- eclipse
 - spring tool suite
- myeclipse - 做很多插件
- idea

03、使用**idea**搭建**ssm**项目

03-01、目的

让大家体验springboot的这个技术好处和优点。解决了什么问题

03-02、创建一个**maven**项目



New Project

Name:

kss-web-ssm

1: 项目名称

Location:

G:\tours\05、学相伴旅游项目实战-高级工程师进阶之路\启动SpringBoot的探索 and 实战\kss-web-ssm

2: 项目目录

▼ Artifact Coordinates

GroupId:

com.kuangstudy.ssm

The name of the artifact group, usually a company domain

ArtifactId:

kss-web-ssm

3: 项目的骨架

The name of the artifact within the group, usually a project name

Version:

1.0-SNAPSHOT

4: 点击下一步

Previous

Next

Cancel

Help

New Project

Maven home path:

G:/tools/apache-maven-3.6.3

1: 配置本地maven

(Version: 3.6.3)

User settings file:

G:\tools\apache-maven-3.6.3\conf\settings.xml

2: 选择你本地的maven配置文件进行指定

☒ Override

Local repository:

F:\respository

☐ Override

Properties

+	-	✎
groupId	com.kuangstudy.ssm	
artifactId	kss-web-ssm	
version	1.0-SNAPSHOT	
archetypeGroupId	org.apache.maven.archetypes	
archetypeArtifactId	maven-archetype-webapp	
archetypeVersion	RELEASE	

3: 点击完成即可

Previous

Finish

Cancel

Help

1: 项目名称

2: 用编写业务代码

3: 资源目录, 用存放xml配置文件

4: 说明页面webapp, 请注意: 才说明是视图页面

5: pom.xml是maven骨架文件

6: web.xml文件是tomcat容器的核心配置文件

刷新: 如果你未来在pom.xml添加了依赖, 你一定要没事多刷新一下

当然你可以执行: **Ctrl + F9**

项目编译, 打包

项目依赖的jar包

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <project xmlns="http://maven.apache.org/POM/4.0.0"
4       xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
5     <modelVersion>4.0.0</modelVersion>
6     <groupId>com.kuangstudy.ssm</groupId>
7     <artifactId>kss-web-ssm</artifactId>
8     <version>1.0-SNAPSHOT</version>
9     <packaging>war</packaging>
10
11     <name>kss-web-ssm Maven Webapp</name>
12     <!-- FIXME change it to the project's website -->
13     <url>http://www.example.com</url>
14
15     <properties>
16         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
17         <maven.compiler.source>1.7</maven.compiler.source>
18     </properties>
  
```

🐱 03-03、项目整合SSM如下

完整项目骨架如下:

完整的项目骨架

```

main
├── java
│   └── com.shsxt
│       ├── base
│       │   ├── AssertUtil
│       │   ├── BaseMapper
│       │   ├── BaseQuery
│       │   ├── BaseService
│       │   └── ParamException
│       ├── controller
│       │   ├── UserController
│       │   │   ├── queryUser(Integer):User
│       │   │   └── userService:UserService
│       ├── dao
│       │   ├── UserMapper
│       │   │   ├── deleteByPrimaryKey(Integer):int
│       │   │   ├── insertSelective(User):int
│       │   │   ├── updateByPrimaryKey(User):int
│       │   │   └── updateByPrimaryKeySelective(User):int
│       ├── mapper
│       │   └── UserMapper.xml
│       ├── po
│       ├── service
│       ├── resources
│       │   ├── db.properties
│       │   ├── generatorConfig.xml
│       │   ├── log4j.properties
│       │   ├── mybatis.xml
│       │   ├── servlet-context.xml
│       │   └── spring.xml
│       ├── webapp
│       │   ├── WEB-INF
│       │   │   ├── web.xml
│       │   └── index.jsp
│       └── pom.xml
  
```

1、在pom.xml文件ssm的相关jar包，如下：

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project
  xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance"
3    xsi:schemaLocation="http://maven.apache.org/PO
  M/4.0.0 http://maven.apache.org/xsd/maven-
    4.0.0.xsd">
4    <modelVersion>4.0.0</modelVersion>
5    <groupId>com.kuangstudy.ssm</groupId>
6    <artifactId>kss-web-ssm</artifactId>
7    <version>1.0-SNAPSHOT</version>
8    <packaging>war</packaging>
9
10   <name>kss-web-ssm Maven Webapp</name>
11   <!-- FIXME change it to the project's website
    -->
12   <url>http://www.example.com</url>
13
14   <properties>
15     <project.build.sourceEncoding>UTF-
16     8</project.build.sourceEncoding>
17
18     <maven.compiler.source>1.8</maven.compiler.sour
    ce>
```

```
17     <maven.compiler.target>1.8</maven.compiler.target>
18 </properties>
19
20 <dependencies>
21     <dependency>
22         <groupId>junit</groupId>
23         <artifactId>junit</artifactId>
24         <version>4.12</version>
25         <scope>test</scope>
26     </dependency>
27     <!-- spring 核心jar -->
28     <dependency>
29         <groupId>org.springframework</groupId>
30         <artifactId>spring-context</artifactId>
31         <version>4.3.2.RELEASE</version>
32     </dependency>
33     <!-- spring 测试jar -->
34     <dependency>
35         <groupId>org.springframework</groupId>
36         <artifactId>spring-test</artifactId>
37         <version>4.3.2.RELEASE</version>
38     </dependency>
39     <!-- spring jdbc -->
40     <dependency>
41         <groupId>org.springframework</groupId>
42         <artifactId>spring-jdbc</artifactId>
43         <version>4.3.2.RELEASE</version>
44     </dependency>
45     <!-- spring事物 -->
```



```
46 <dependency>
47     <groupId>org.springframework</groupId>
48     <artifactId>spring-tx</artifactId>
49     <version>4.3.2.RELEASE</version>
50 </dependency>
51 <!-- aspectj切面编程的jar -->
52 <dependency>
53     <groupId>org.aspectj</groupId>
54     <artifactId>aspectjweaver</artifactId>
55     <version>1.8.9</version>
56 </dependency>
57 <!-- c3p0 连接池 -->
58 <dependency>
59     <groupId>c3p0</groupId>
60     <artifactId>c3p0</artifactId>
61     <version>0.9.1.2</version>
62 </dependency>
63 <!-- mybatis -->
64 <dependency>
65     <groupId>org.mybatis</groupId>
66     <artifactId>mybatis</artifactId>
67     <version>3.4.1</version>
68 </dependency>
69 <!-- 添加mybatis与Spring整合的核心包 -->
70 <dependency>
71     <groupId>org.mybatis</groupId>
72     <artifactId>mybatis-spring</artifactId>
73     <version>1.3.0</version>
74 </dependency>
75 <!-- mysql 驱动包 -->
76 <dependency>
```

```
77         <groupId>mysql</groupId>
78         <artifactId>mysql-connector-
java</artifactId>
79         <version>8.0.25</version>
80     </dependency>
81     <!-- 日志打印相关的jar -->
82     <dependency>
83         <groupId>org.slf4j</groupId>
84         <artifactId>slf4j-log4j12</artifactId>
85         <version>1.7.2</version>
86     </dependency>
87     <dependency>
88         <groupId>org.slf4j</groupId>
89         <artifactId>slf4j-api</artifactId>
90         <version>1.7.2</version>
91     </dependency>
92     <!-- spring web -->
93     <dependency>
94         <groupId>org.springframework</groupId>
95         <artifactId>spring-web</artifactId>
96         <version>4.3.2.RELEASE</version>
97     </dependency>
98     <!-- spring mvc -->
99     <dependency>
100         <groupId>org.springframework</groupId>
101         <artifactId>spring-webmvc</artifactId>
102         <version>4.3.2.RELEASE</version>
103     </dependency>
104     <!-- web servlet -->
105     <dependency>
106         <groupId>javax.servlet</groupId>
```

```
107         <artifactId>javax.servlet-api</artifactId>
108         <version>3.0.1</version>
109         <scope>provided</scope>
110     </dependency>
111     <!-- 添加json 依赖jar包 -->
112     <dependency>
113
114         <groupId>com.fasterxml.jackson.core</groupId>
115         <artifactId>jackson-core</artifactId>
116         <version>2.7.0</version>
117     </dependency>
118
119     <dependency>
120         <groupId>com.fasterxml.jackson.core</groupId>
121         <artifactId>jackson-databind</artifactId>
122         <version>2.7.0</version>
123     </dependency>
124
125     <dependency>
126         <groupId>com.fasterxml.jackson.core</groupId>
127         <artifactId>jackson-
128         annotations</artifactId>
129         <version>2.7.0</version>
130     </dependency>
131     <!-- 分页插件配置 -->
132     <dependency>
133         <groupId>com.github.pagehelper</groupId>
134         <artifactId>pagehelper</artifactId>
135         <version>4.1.0</version>
136     </dependency>
```

```
134
135     <!-- 文件上传包依赖 -->
136     <dependency>
137         <groupId>commons-fileupload</groupId>
138         <artifactId>commons-
fileupload</artifactId>
139         <version>1.3.2</version>
140     </dependency>
141 </dependencies>
142
143 <build>
144     <finalName>kss-web-ssm</finalName>
145     <pluginManagement><!-- lock down plugins
versions to avoid using Maven defaults (may be
moved to parent pom) -->
146         <plugins>
147             <plugin>
148                 <artifactId>maven-clean-
plugin</artifactId>
149                 <version>3.1.0</version>
150             </plugin>
151             <!-- see
http://maven.apache.org/ref/current/maven-
core/default-
bindings.html#Plugin\_bindings\_for\_war\_packaging
-->
152             <plugin>
153                 <artifactId>maven-resources-
plugin</artifactId>
154                 <version>3.0.2</version>
155             </plugin>
```

```
156         <plugin>
157             <artifactId>maven-compiler-
plugin</artifactId>
158             <version>3.8.0</version>
159         </plugin>
160         <plugin>
161             <artifactId>maven-surefire-
plugin</artifactId>
162             <version>2.22.1</version>
163         </plugin>
164         <plugin>
165             <artifactId>maven-war-
plugin</artifactId>
166             <version>3.2.2</version>
167         </plugin>
168         <plugin>
169             <artifactId>maven-install-
plugin</artifactId>
170             <version>2.5.2</version>
171         </plugin>
172         <plugin>
173             <artifactId>maven-deploy-
plugin</artifactId>
174             <version>2.8.2</version>
175         </plugin>
176     </plugins>
177 </pluginManagement>
178 </build>
179 </project>
```

用springboot来做的话,只需要如下就解决上面所有的依赖问题，如下：

```
1 <!-- 日志打印相关的jar -->
2 <dependency>
3     <groupId>org.springframework</groupId>
4     <artifactId>spring-boot-starter-
web</artifactId>
5 </dependency>
```

本质上没变化：因为spring-boot-starter-web 内部会自动的去帮你去管理上面的这些包。进行下载配置。

2、配置spring的配置文件

spring.xml：这个文件的作用其实就去整合mybatis和数据源和事务等。如下：

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans
  xmlns="http://www.springframework.org/schema/beans"
3
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
  xmlns:context="http://www.springframework.org/sch
ema/context"
```

4

```
xmlns:aop="http://www.springframework.org/schema/aop"
```

```
xmlns:tx="http://www.springframework.org/schema/tx"
```

5

```
xmlns:schemaLocation="http://www.springframework.org/schema/beans
```

6

```
http://www.springframework.org/schema/beans/spring-beans.xsd
```

7

```
http://www.springframework.org/schema/context
```

8

```
http://www.springframework.org/schema/context/spring-context.xsd
```

9

```
http://www.springframework.org/schema/aop
```

10

```
http://www.springframework.org/schema/aop/spring-aop.xsd
```

11

```
http://www.springframework.org/schema/tx
```

12

```
http://www.springframework.org/schema/tx/spring-tx-3.0.xsd">
```

13

```
<!-- 扫描基本包 过滤controller层 -->
```

14

```
<context:component-scan base-
```

15

```
package="com.shsxt" >
```

16

```
<context:exclude-filter type="annotation" expression="org.springframework.stereotype.Controller" />
```

```
17     </context:component-scan>
18
19     <!-- 加载properties 配置文件 -->
20     <context:property-placeholder
location="classpath:db.properties" />
21
22     <aop:aspectj-autoproxy /><!-- aop -->
23
24     <!-- 配置c3p0 数据源 -->
25     <bean id="dataSource"
class="com.mchange.v2.c3p0.ComboPooledDataSource"
>
26         <property name="driverClass"
value="${jdbc.driver}"></property>
27         <property name="jdbcUrl"
value="${jdbc.url}"></property>
28         <property name="user"
value="${jdbc.username}"></property>
29         <property name="password"
value="${jdbc.password}"></property>
30     </bean>
31
32     <!-- 配置事务管理器 -->
33     <bean id="txManager"
34
class="org.springframework.jdbc.datasource.DataS
ourceTransactionManager">
35         <property name="dataSource"
ref="dataSource"></property>
36     </bean>
37
```



```
38      <!-- 设置事物增强 -->
39      <tx:advice id="txAdvice" transaction-
manager="txManager">
40          <tx:attributes>
41              <tx:method name="get*" read-
only="true" />
42              <tx:method name="find*" read-
only="true" />
43              <tx:method name="query*" read-
only="true" />
44              <tx:method name="load*" read-
only="true" />
45              <tx:method name="add*"
propagation="REQUIRED" />
46              <tx:method name="insert*"
propagation="REQUIRED" />
47              <tx:method name="update*"
propagation="REQUIRED" />
48              <tx:method name="delete*"
propagation="REQUIRED" />
49          </tx:attributes>
50      </tx:advice>
51
52      <!-- aop 切面配置 -->
53      <aop:config>
54          <aop:pointcut id="servicePointcut"
expression="execution(*
com.shsxt.service..*.*(..))" />
55          <aop:advisor advice-ref="txAdvice"
pointcut-ref="servicePointcut" />
56      </aop:config>
```

```

58
59     <!-- 配置 sqlSessionFactory -->
60     <bean id="sqlSessionFactory"
61         class="org.mybatis.spring.SqlSessionFactoryBean">
62         <property name="dataSource"
63             ref="dataSource"></property>
64         <property name="configLocation"
65             value="classpath:mybatis.xml" />
66         <property name="mapperLocations"
67             value="classpath:com/shsxt/mapper/*.xml" />
68     </bean>
69     <!-- 配置扫描器 -->
70     <bean id="mapperScanner"
71         class="org.mybatis.spring.mapper.MapperScannerCon
72         figurer">
73         <!-- 扫描com.shsxt.dao这个包以及它的子包下的所
74             有映射接口类 -->
75         <property name="basePackage"
76             value="com.shsxt.dao" />
77         <property
78             name="sqlSessionFactoryBeanName"
79             value="sqlSessionFactory" />
80     </bean>
81 </beans>

```

3、配置springmvc配置文件：

servlet-context.xml：配置springmvc,包括：json处理，拦截器，视图转发等如下：

```

1 <?xml version="1.0" encoding="UTF-8"?>

```

```
2 <beans
  xmlns="http://www.springframework.org/schema/beans"
3
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance"
4
  xmlns:mvc="http://www.springframework.org/schema/
  mvc"
5
  xmlns:context="http://www.springframework.org/sch
  ema/context"
6
  xmlns:aop="http://www.springframework.org/schema/
  aop"
7
  xmlns:tx="http://www.springframework.org/schema/t
  x"
8      xsi:schemaLocation="
9          http://www.springframework.org/schema/mvc
10             http://www.springframework.org/schema/mvc/spring
             -mvc-3.2.xsd
11             http://www.springframework.org/schema/beans
12             http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
13             http://www.springframework.org/schema/context
```

14 <http://www.springframework.org/schema/context/spring-context-3.0.xsd>

15 <http://www.springframework.org/schema/aop>

16 <http://www.springframework.org/schema/aop/spring-aop-2.5.xsd>

17 <http://www.springframework.org/schema/tx>

18 [http://www.springframework.org/schema/tx/spring-tx-2.5.xsd">](http://www.springframework.org/schema/tx/spring-tx-2.5.xsd)

19
20
21 <!-- 扫描com.shsxt.controller 下包 -->

22 <context:component-scan base-
package="com.shsxt.controller" />

23 <!-- mvc 请求映射处理器与适配器 -->

24 <mvc:annotation-driven />

25
26 <!--配置视图解析器 默认的视图解析器- -->

27 <bean id="defaultViewResolver"

28 class="org.springframework.web.servlet.view.InternalResourceViewResolver">

29 <property name="viewClass"

30 value="org.springframework.web.servlet.view.Jstl
view" />

31 <property name="contentType"
value="text/html" />

```
32         <property name="prefix" value="/WEB-INF/jsp/" />
33         <property name="suffix" value=".jsp" />
34     </bean>
35
36
37     <!-- json 支持 -->
38     <bean
39
40         class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping">
41     </bean>
42     <bean
43
44         class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter">
45         <property name="messageConverters">
46             <list>
47                 <bean
48
49                     class="org.springframework.http.converter.json.MappingJackson2HttpMessageConverter" />
50             </list>
51         </property>
52     </bean>
53
54     <!-- 文件上传配置 -->
55     <bean id="multipartResolver"
```

```

54         class="org.springframework.web.multipart.commons
        .CommonsMultipartResolver">
55             <property name="maxUploadSize">
56                 <value>104857600</value>
57             </property>
58             <property name="maxInMemorySize">
59                 <value>4096</value>
60             </property>
61         </bean>
62
63 </beans>

```

4、web.xml配置

把spring.xml和springmvc-context.xml配置到web.xml中

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   instance"
3
   xmlns="http://java.sun.com/xml/ns/javaee"
4
   xsi:schemaLocation="http://java.sun.com/xml/ns/ja
   vae http://java.sun.com/xml/ns/javaee/web-
   app_3_0.xsd"
5       id="WebApp_ID" version="3.0">
6     <context-param>
7       <param-name>contextConfigLocation</param-
       name>

```

```
8         <param-value>classpath:spring.xml</param-  
value>  
9     </context-param>  
10    <listener>  
11        <listener-  
class>org.springframework.web.context.ContextLoad  
erListener</listener-class>  
12    </listener>  
13    <filter>  
14        <description>char encoding  
filter</description>  
15        <filter-name>encodingFilter</filter-name>  
16        <filter-  
class>org.springframework.web.filter.CharacterEnc  
odingFilter</filter-class>  
17        <init-param>  
18            <param-name>encoding</param-name>  
19            <param-value>UTF-8</param-value>  
20        </init-param>  
21    </filter>  
22    <filter-mapping>  
23        <filter-name>encodingFilter</filter-name>  
24        <url-pattern>/*</url-pattern>  
25    </filter-mapping>  
26    <servlet>  
27        <servlet-name>springMvc</servlet-name>  
28        <servlet-  
class>org.springframework.web.servlet.Dispatchers  
ervlet</servlet-class>  
29        <init-param>
```

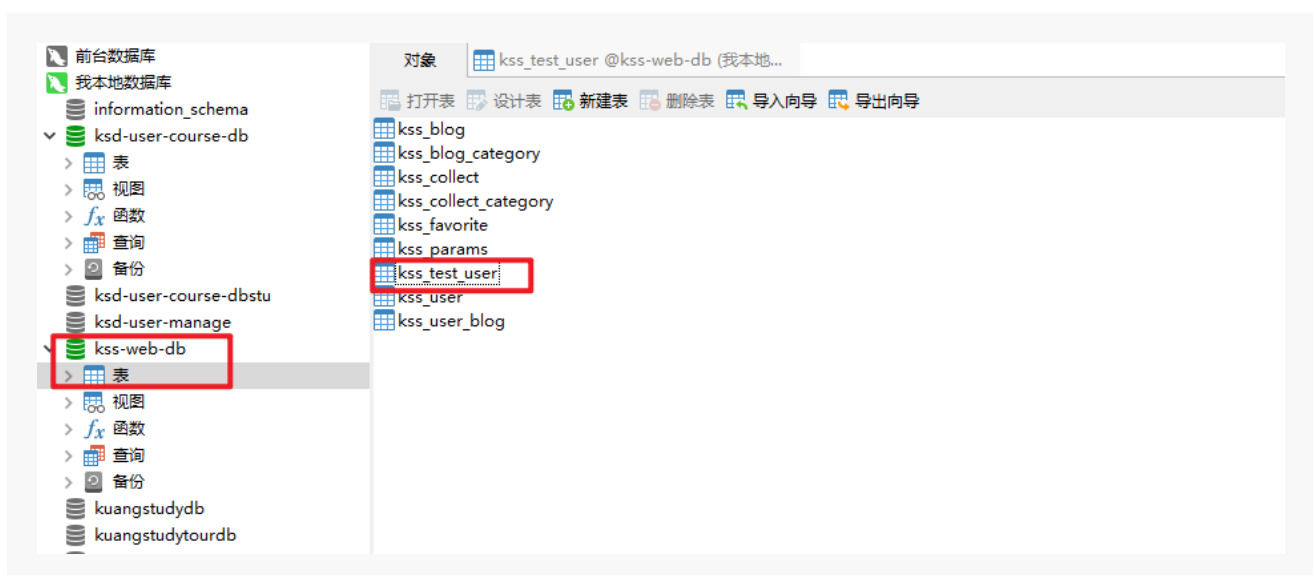
```

30         <param-
name>contextConfigLocation</param-name>
31         <param-value>classpath:servlet-
context.xml</param-value>
32     </init-param>
33     <load-on-startup>1</load-on-startup>
34 </servlet>
35 <servlet-mapping>
36     <servlet-name>springMvc</servlet-name>
37     <url-pattern>/</url-pattern>
38 </servlet-mapping>
39 </web-app>

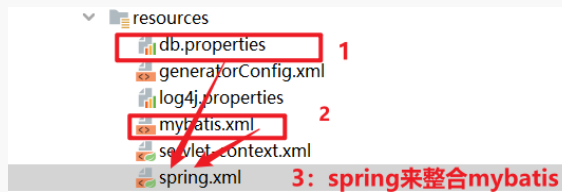
```

5、整合mybatis配置-- 配置数据源属性配置

前提：准备数据库kss-web-db 和一个数据库表： kss_test_user



配置文件结构如下：



5-1、db.properties 如下:

```
1 jdbc.driver=com.mysql.jdbc.Driver
2 jdbc.url=jdbc:mysql://localhost:3306/kss-web-db?
  useUnicode=true&characterEncoding=utf8
3 jdbc.username=root
4 jdbc.password=mkxiaoer
```

5-2、mybatis.xml如下:

```
1 <!DOCTYPE configuration
2     PUBLIC "-//mybatis.org//DTD Config
3     3.0//EN"
4     "http://mybatis.org/dtd/mybatis-3-
5     config.dtd">
6
7 <configuration>
8     <!-- po 包扫描 -->
9     <typeAliases>
10         <package name="com.shsxt.po" />
11     </typeAliases>
12     <plugins>
13         <!-- com.github.pagehelper为PageHelper类所
14         在包名 -->
15         <plugin
16             interceptor="com.github.pagehelper.PageHelper">
```

```
13         <property name="dialect"
value="mysql" />
14         <!-- 该参数默认为false -->
15         <!-- 设置为true时，会将RowBounds第一个参
数offset当成pageNum页码使用 -->
16         <!-- 和startPage中的pageNum效果一样 -->
17         <property name="offsetAsPageNum"
value="true" />
18         <!-- 该参数默认为false -->
19         <!-- 设置为true时，使用RowBounds分页会进
行count查询 -->
20         <property name="rowBoundswithCount"
value="true" />
21         <!-- 设置为true时，如果pageSize=0或者
RowBounds.limit = 0就会查询出全部的结果 -->
22         <!-- （相当于没有执行分页查询，但是返回结果
仍然是Page类型） -->
23         <property name="pageSizeZero"
value="true" />
24         <!-- 3.3.0版本可用 - 分页参数合理化，默认
false禁用 -->
25         <!-- 启用合理化时，如果pageNum<1会查询第一
页，如果pageNum>pages会查询最后一页 -->
26         <!-- 禁用合理化时，如果pageNum<1或
pageNum>pages会返回空数据 -->
27         <property name="reasonable"
value="true" />
28         <!-- 3.5.0版本可用 - 为了支持
startPage(Object params)方法 -->
29         <!-- 增加了一个`params`参数来配置参数映
射，用于从Map或ServletRequest中取值 -->
```

```

30         <!-- 可以配置
        pageNum,pageSize,count,pageSizeZero,reasonable,不
        配置映射的用默认值 -->
31         <property name="params"
32             value="pageNum=start;pageSize=limit;pageSizeZero
            =zero;reasonable=hel;count=countsql" />
33     </plugin>
34 </plugins>
35 </configuration>

```

5-3、新建一个实体bean

比如kss_test_user

```

1 package com.shsxt.po;
2
3 public class User {
4     private Integer id;
5
6     private String userName;
7
8     private String userPwd;
9
10    private String realName;
11
12    private String nation;
13
14    private Integer cardId;
15
16    public Integer getId() {

```

```
17         return id;
18     }
19
20     public void setId(Integer id) {
21         this.id = id;
22     }
23
24     public String getUserName() {
25         return userName;
26     }
27
28     public void setUserName(String userName) {
29         this.userName = userName == null ? null :
userName.trim();
30     }
31
32     public String getUserPwd() {
33         return userPwd;
34     }
35
36     public void setUserPwd(String userPwd) {
37         this.userPwd = userPwd == null ? null :
userPwd.trim();
38     }
39
40     public String getRealName() {
41         return realName;
42     }
43
44     public void setRealName(String realName) {
```

```
45         this.realName = realName == null ? null :
realName.trim();
46     }
47
48     public String getNation() {
49         return nation;
50     }
51
52     public void setNation(String nation) {
53         this.nation = nation == null ? null :
nation.trim();
54     }
55
56     public Integer getCardId() {
57         return cardId;
58     }
59
60     public void setCardId(Integer cardId) {
61         this.cardId = cardId;
62     }
63 }
```

5-4、mapper和dao新建

UserMapper.java

```

1 package com.shsxt.dao;
2
3 import com.shsxt.base.BaseMapper;
4 import com.shsxt.po.User;
5
6 public interface UserMapper extends
    BaseMapper<User> {
7     int deleteByPrimaryKey(Integer id);
8     int insertSelective(User record);
9     int updateByPrimaryKeySelective(User record);
10    int updateByPrimaryKey(User record);
11 }

```

UserMapper.xml

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD
    Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-
    3-mapper.dtd" >
3 <mapper namespace="com.shsxt.dao.UserMapper" >
4     <resultMap id="BaseResultMap"
        type="com.shsxt.po.User" >
5         <id column="id" property="id"
            jdbcType="INTEGER" />
6         <result column="user_name"
            property="userName" jdbcType="VARCHAR" />
7         <result column="user_pwd" property="userPwd"
            jdbcType="VARCHAR" />
8         <result column="real_name"
            property="realName" jdbcType="VARCHAR" />

```

```
9      <result column="nation" property="nation"
jdbcTemplate="VARCHAR" />
10      <result column="card_id" property="cardId"
jdbcTemplate="INTEGER" />
11  </resultMap>
12  <sql id="Base_Column_List" >
13      id, user_name, user_pwd, real_name, nation,
card_id
14  </sql>
15  <select id="queryById"
resultMap="BaseResultMap"
parameterType="java.lang.Integer" >
16      select
17      <include refid="Base_Column_List" />
18      from kss_test_user
19      where id = #{id,jdbcTemplate=INTEGER}
20  </select>
21  <delete id="deleteByPrimaryKey"
parameterType="java.lang.Integer" >
22      delete from kss_test_user
23      where id = #{id,jdbcTemplate=INTEGER}
24  </delete>
25  <insert id="insert"
parameterType="com.shsxt.po.User" >
26      insert into kss_test_user (id, user_name,
user_pwd,
27      real_name, nation, card_id
28      )
29      values (#{id,jdbcTemplate=INTEGER}, #
{userName,jdbcTemplate=VARCHAR}, #
{userPwd,jdbcTemplate=VARCHAR},
```

```
30      #{realName,jdbcType=VARCHAR}, #
      {nation,jdbcType=VARCHAR}, #
      {cardId,jdbcType=INTEGER}
31    )
32  </insert>
33  <insert id="insertSelective"
parameterType="com.shsxt.po.User" >
34    insert into kss_test_user
35    <trim prefix="(" suffix=")"
suffixOverrides=", " >
36      <if test="id != null" >
37        id,
38      </if>
39      <if test="userName != null" >
40        user_name,
41      </if>
42      <if test="userPwd != null" >
43        user_pwd,
44      </if>
45      <if test="realName != null" >
46        real_name,
47      </if>
48      <if test="nation != null" >
49        nation,
50      </if>
51      <if test="cardId != null" >
52        card_id,
53      </if>
54    </trim>
55    <trim prefix="values (" suffix=")"
suffixOverrides=", " >
```



```
56      <if test="id != null" >
57          #{id,jdbcType=INTEGER},
58      </if>
59      <if test="userName != null" >
60          #{userName,jdbcType=VARCHAR},
61      </if>
62      <if test="userPwd != null" >
63          #{userPwd,jdbcType=VARCHAR},
64      </if>
65      <if test="realName != null" >
66          #{realName,jdbcType=VARCHAR},
67      </if>
68      <if test="nation != null" >
69          #{nation,jdbcType=VARCHAR},
70      </if>
71      <if test="cardId != null" >
72          #{cardId,jdbcType=INTEGER},
73      </if>
74  </trim>
75 </insert>
76 <update id="updateByPrimaryKeySelective"
parameterType="com.shsxt.po.User" >
77     update kss_test_user
78     <set >
79         <if test="userName != null" >
80             user_name = #
{userName,jdbcType=VARCHAR},
81         </if>
82         <if test="userPwd != null" >
83             user_pwd = #{userPwd,jdbcType=VARCHAR},
84         </if>
```

```

85         <if test="realName != null" >
86             real_name = #
            {realName,jdbcType=VARCHAR},
87         </if>
88         <if test="nation != null" >
89             nation = #{nation,jdbcType=VARCHAR},
90         </if>
91         <if test="cardId != null" >
92             card_id = #{cardId,jdbcType=INTEGER},
93         </if>
94     </set>
95     where id = #{id,jdbcType=INTEGER}
96 </update>
97 <update id="updateByPrimaryKey"
parameterType="com.shsxt.po.User" >
98     update kss_test_user
99     set user_name = #
        {userName,jdbcType=VARCHAR},
100     user_pwd = #{userPwd,jdbcType=VARCHAR},
101     real_name = #{realName,jdbcType=VARCHAR},
102     nation = #{nation,jdbcType=VARCHAR},
103     card_id = #{cardId,jdbcType=INTEGER}
104     where id = #{id,jdbcType=INTEGER}
105 </update>
106 </mapper>

```

5-5、新建一个UserService

```

1 package com.shsxt.service;

```

```
2
3 import com.shsxt.base.BaseService;
4 import com.shsxt.dao.UserMapper;
5 import com.shsxt.po.User;
6 import org.springframework.stereotype.Service;
7
8 import javax.annotation.Resource;
9 @Service
10 public class UserService extends
    BaseService<User> {
11     @Resource
12     private UserMapper userMapper;
13 }
14
```

5-6、新建一个UserController

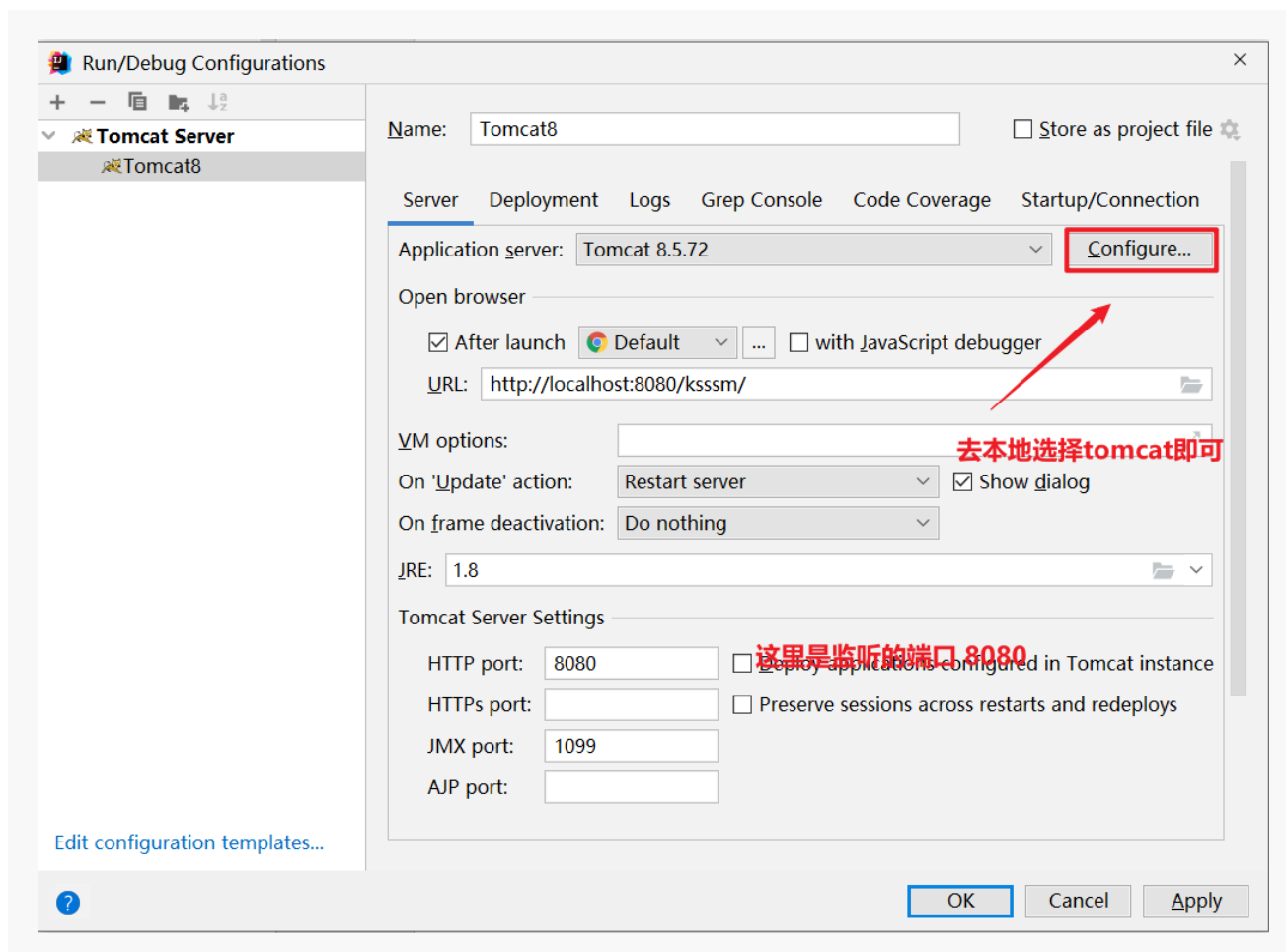
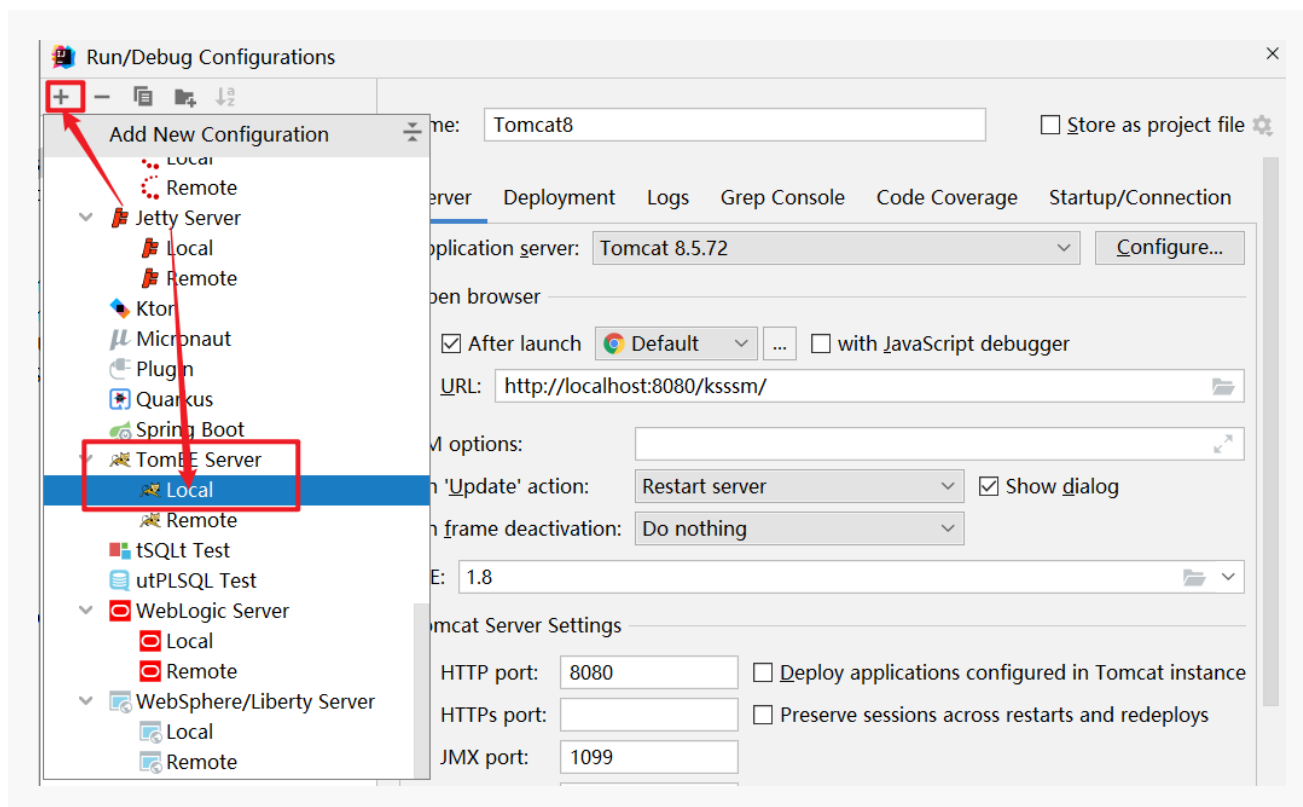
```
1 package com.shsxt.controller;
2
3 import com.shsxt.po.User;
4 import com.shsxt.service.UserService;
5 import org.springframework.stereotype.Controller;
6 import
    org.springframework.web.bind.annotation.RequestMa
    pping;
7 import
    org.springframework.web.bind.annotation.ResponseB
    ody;
8
```

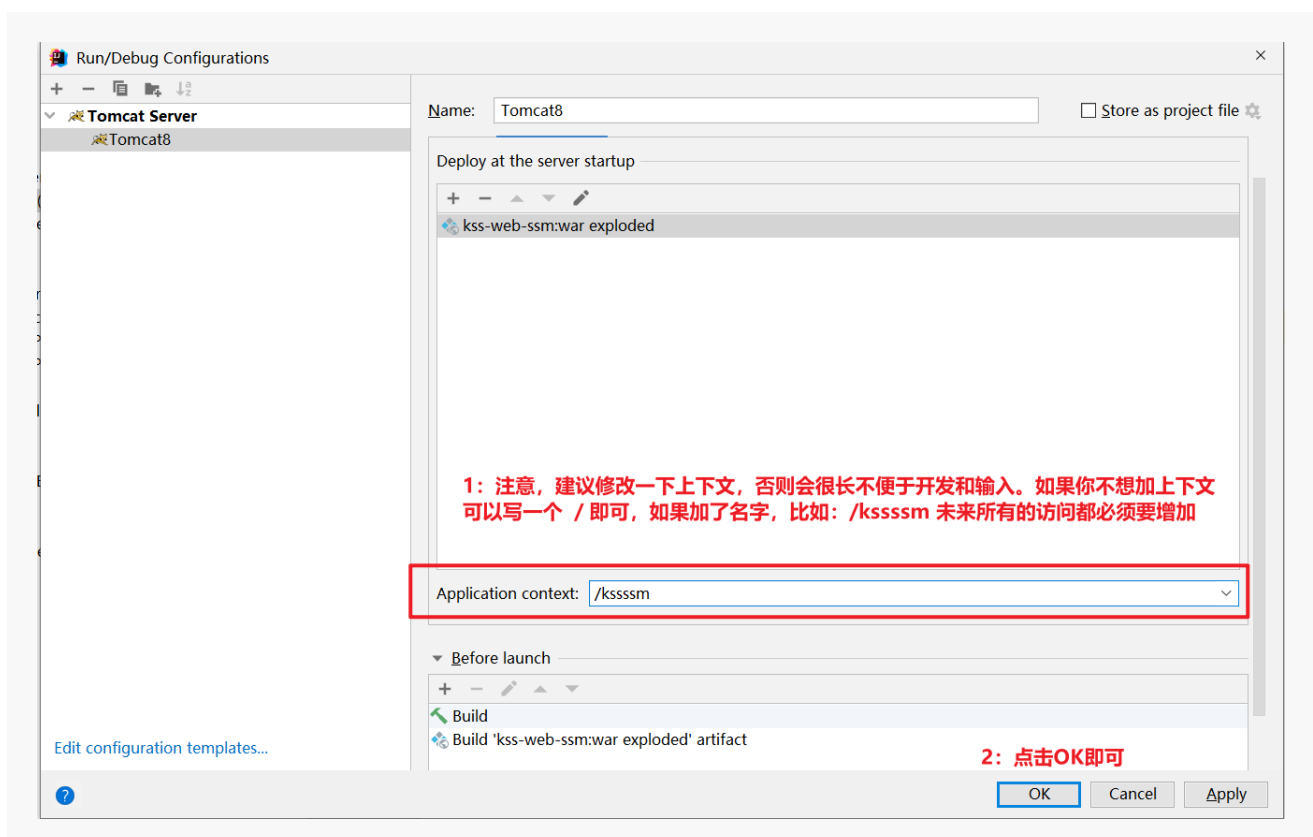
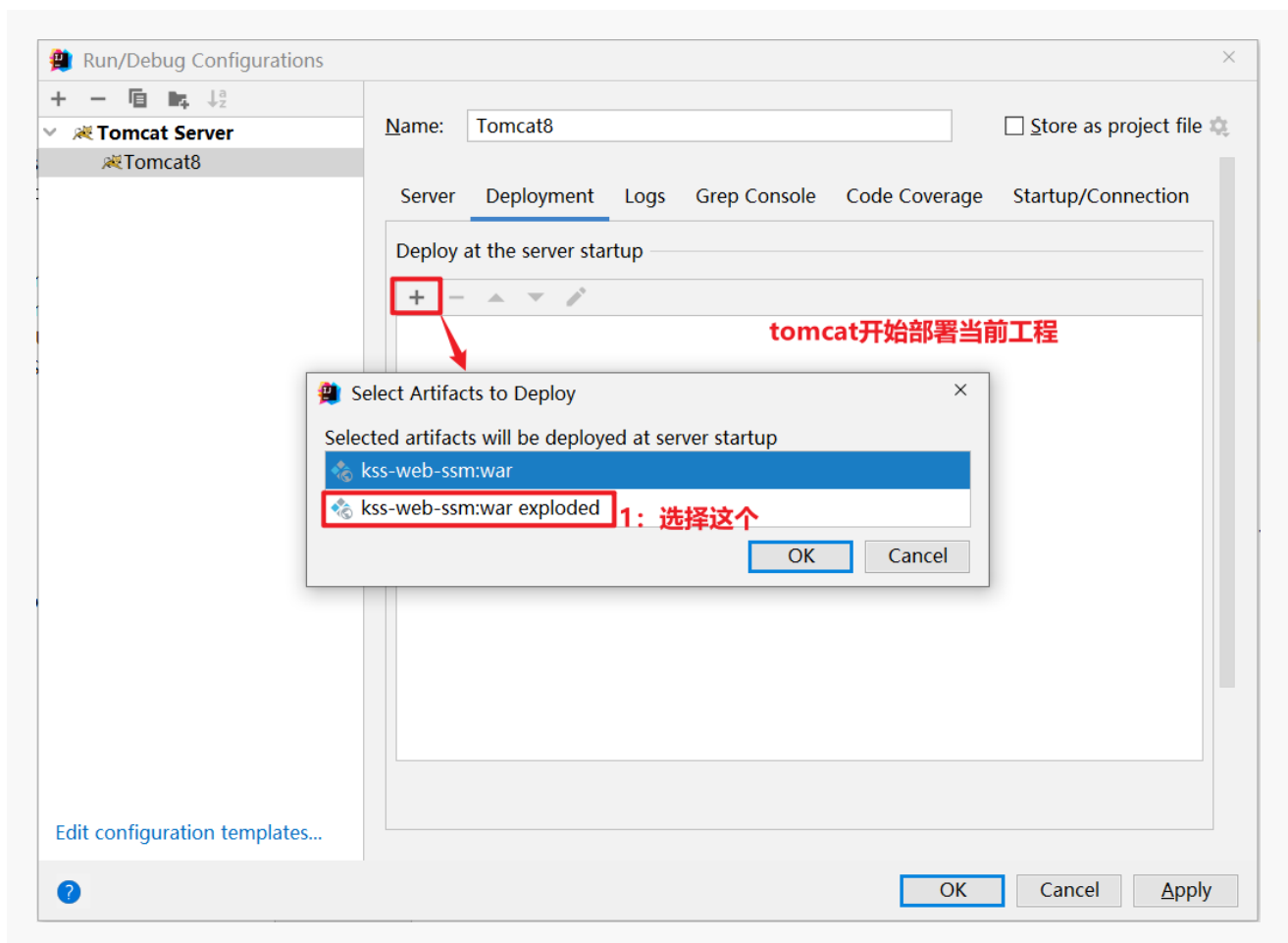
```
9  import javax.annotation.Resource;
10
11  @Controller
12  @RequestMapping("user")
13  public class UserController {
14      @Resource
15      private UserService userService;
16
17
18      @RequestMapping("queryUser")
19      @ResponseBody
20      public User queryUser(Integer id) throws
Exception {
21          return userService.queryById(id);
22      }
23  }
24  -
```

5-7、未来每天的工作流程如下：

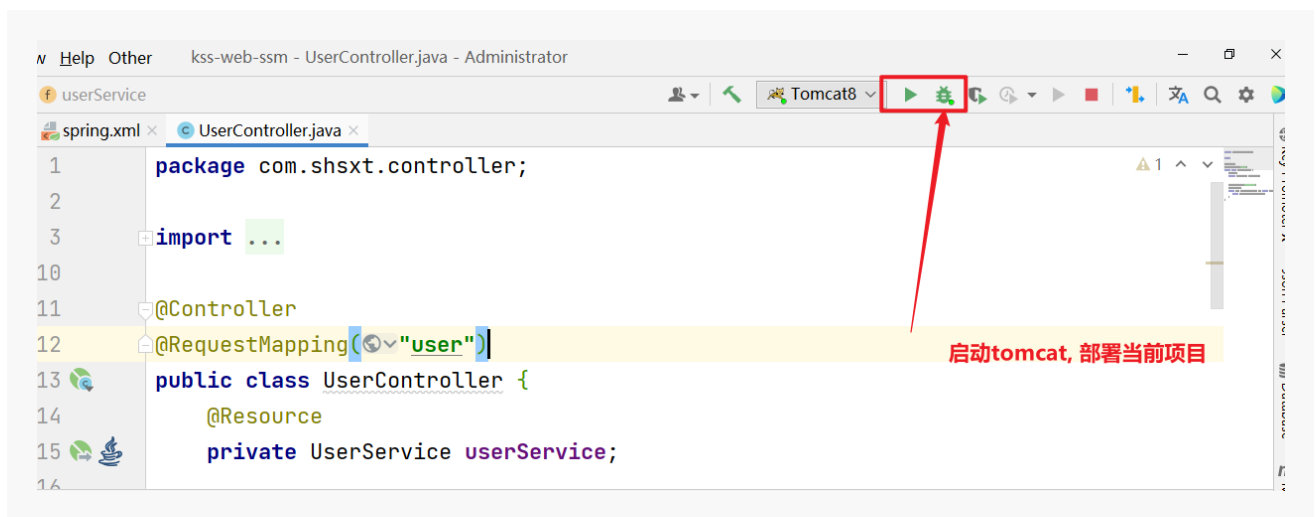
整体步骤如下：

2、选择配置本地的tomcat





部署项目：



然后在浏览器访问：<http://localhost:8080/ksssm/user/queryUser?id=1>，如果能够查询出来：整合ssm成功：

