

项目模块代码组织

ioc模块

xml配置版

student

```
<bean id="student1" class="com.soft1851.spring.Student">
    <property name="name" value="王锋"/>
    <property name="phone" ref="phone1"/>
</bean>
<bean id="student2"
class="com.soft1851.spring.Student">
    <constructor-arg name="name" value="颜子浩"/>
    <constructor-arg name="phone" ref="phone2"/>
</bean>
<bean id="student3"
class="com.soft1851.spring.Student">
    <property name="name" value="刘恋"/>
    <property name="phone" ref="phone3"/>
</bean>
```

Phone

```
<bean id="phone1" class="com.soft1851.spring.Phone">
    <constructor-arg name="brand" value="iPhone11"/>
    <constructor-arg name="price" value="8888.8"/>
</bean>
<bean id="phone2" class="com.soft1851.spring.Phone">
<property name="brand" value="iPhoneX"/>
<property name="price" value="6666.6"/>
</bean>
<bean id="phone3" class="com.soft1851.spring.Phone">
    <constructor-arg name="brand" value="iPhoneXmax"/>
    <constructor-arg name="price" value="1111.1"/>
</bean>
```

排序

```
<bean id="bubbleSort" class="com.spring.BubbleSort"/>
```

list、set、map

```
<bean id="student"
class="com.soft1851.ioc.entity.Student">
    <property name="name" value="xgp" />
    <property name="phone" ref="phone" />
    <property name="phones">
        <list>
            <ref bean="phone" />
        </list>
    </property>
</bean>

    <bean id="phone" class="com.soft1851.ioc.entity.Phone"
p:name="iPhoneX" p:price="9999" />
    <bean id="phone2"
class="com.soft1851.ioc.entity.Phone" p:name="iPhoneXmax"
p:price="11111" />
    <bean id="test" class="com.soft1851.ioc.entity.Demo">
        <constructor-arg>
            <list>
                <map>
                    <entry key="stu1" value="student1"/>
                    <entry key="stu2" value="student2" />
                </map>
            </list>
        </constructor-arg>
        <constructor-arg>
            <set>
                <ref bean="phone" />
                <ref bean="phone2" />
            </set>
        </constructor-arg>
        <constructor-arg>
            <list>
                <value>stu1</value>
                <value>stu2</value>
            </list>
        </constructor-arg>
    </bean>
```

注解配置版

```
import com.soft1851.ioc.entity.Student;
import org.springframework.context.annotation.Bean;
import
org.springframework.context.annotation.Configuration;
```

```

@Configuration
public class AppConfig {
    @Bean
    public Student student(){
        Student student1 = new Student();
        student1.setName("王锋");
        student1.setPhone("phone1");
        return new Student();

        Student student2 = new Student();
        student1.setName("颜子皓");
        student1.setPhone("phone2");
        return new Student();

        Student student1 = new Student();
        student1.setName("刘恋");
        student1.setPhone("phone3");
        return new Student();
    }
}

```

orm模块

xml配置版

```

<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

        xmlns:context="http://www.springframework.org/schema/conte
xt"

        xsi:schemaLocation="http://www.springframework.org/schema/
beans http://www.springframework.org/schema/beans/spring-
beans.xsd http://www.springframework.org/schema/context
https://www.springframework.org/schema/context/spring-
context.xsd">
    <context:annotation-config/>
    <context:component-scan base-
package="com.soft1851.orm.config"/>
    <context:property-placeholder
location="db.properties"/>
    <bean id="dataSource"
class="com.alibaba.druid.pool.DruidDataSource" init-
method="init">

```

```

        <property name="driverClassName"
value="${jdbc.driverClassName}"/>
        <property name="url" value="${jdbc.url}"/>
        <property name="username"
value="${jdbc.username}"/>
        <property name="password"
value="${jdbc.password}"/>
        <property name="initialSize" value="8"/>
    </bean>
    <bean id="jdbcTemplate"
class="org.springframework.jdbc.core.JdbcTemplate">
        <property name="dataSource" ref="dataSource"/>
    </bean>
</beans>

```

注解配置版

```

import com.alibaba.druid.pool.DruidDataSource;
import org.springframework.context.annotation.Bean;
import
org.springframework.context.annotation.ComponentScan;
import
org.springframework.context.annotation.Configuration;
import org.springframework.jdbc.core.JdbcTemplate;

@Configuration
@ComponentScan("com.soft1851.orm")
public class JdbcConfig {
    @Bean
    public DruidDataSource dataSource() {
        DruidDataSource source = new DruidDataSource();

        source.setDriverClassName("com.mysql.jdbc.Driver");

        source.setUrl("jdbc:mysql://localhost:3306/db_java_basic?
useUnicode=true&characterEncoding=utf8&useSSL=false&autoRe
connect=true");
        source.setUsername("root");
        source.setPassword("root");
        //配置初始化大小、最小、最大
        source.setInitialSize(8);
        source.setMinIdle(1);
        source.setMaxActive(20);
        //配置获取连接等待超时的时间
        source.setMaxWait(60000);
        //配置间隔多久进行一次检测，检测需要关闭的空间连接，单位是毫
秒
    }
}

```

```

        source.setTimeBetweenEvictionRunsMillis(60000);
        //配置一个连接在池中最小生成的时间，单位是毫秒
        source.setMinEvictableIdleTimeMillis(300000);
        //禁止自动提交，实现事务管理
        source.setDefaultAutoCommit(false);
        //设置连接池启用预处理事务集
        source.setPoolPreparedStatements(true);
        return source;
    }

    @Bean
    public JdbcTemplate jdbcTemplate(DruidDataSource
dataSource){
        return new JdbcTemplate(dataSource);
    }
}

```

web模块

```

<!-- 监听tomcat启动-->
<listener>
    <listener-
class>org.springframework.web.context.ContextLoaderListene
r</listener-class>
</listener>

<!--web容器配置，当tomcat启动，加载web.xml配置文件-->
<servlet>
    <servlet-name>springMVC</servlet-name>
    <servlet-
class>org.springframework.web.servlet.DispatcherServlet</s
ervlet-class>
    <init-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>classpath:springMVC-
servlet.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>

<!--mvc控制器映射-->
<servlet-mapping>
    <servlet-name>springMVC</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>

<!--启动mvc注解-->

```

```
<mvc:annotation-driven />
<!--扫描含有注解的包（控制器所在包）-->
<context:component-scan base-
package="com.soft1851.spring.web.controller" />
<!--保证静态资源不被拦截-->
<mvc:default-servlet-handler />
```

```
import
org.springframework.web.bind.annotation.RequestMapping;
import
org.springframework.web.bind.annotation.RestController;

@RestController
public class Helloweb{
    @RequestMapping(value = "/hello")
    public String getHello() {
        return "hello";
    }
}
```