

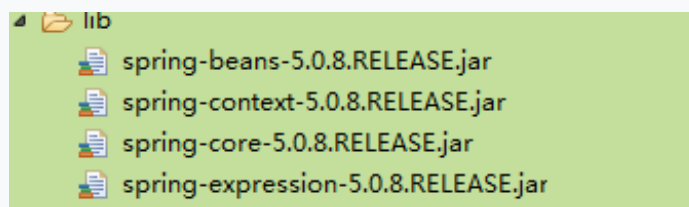
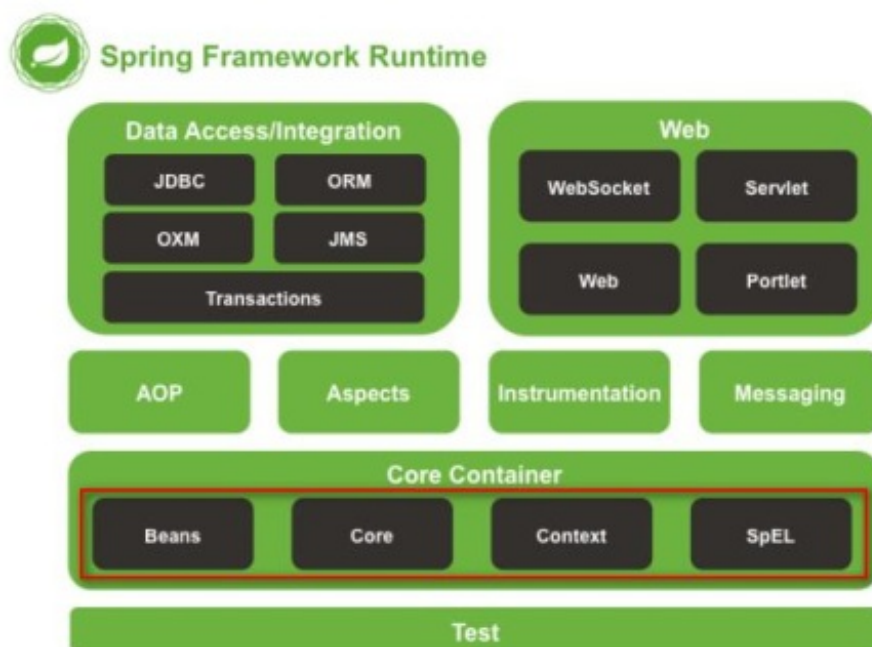
02 Spring_HelloWord

Spring

一：创建一个Web项目,并导入jar包

1、导入jar包

1、核心包



2、日志包(新版本)

spring-framework-3.0.2.RELEASE-dependencies.zip(org.apache.commons\com.springsource.org.apache.commons.logging\1.1.1 - 解包大小为 171.1 MB)				
名称	压缩前	压缩后	类型	修改日期
.. (上级目录)			文件夹	
com.springsource.org.apache.commons.logging-1.1.1.jar	60.0 KB	55.0 KB	JAR 文件	2010-04-02 11:08
com.springsource.org.apache.commons.logging-sources-1.1.1.jar	73.2 KB	71.0 KB	JAR 文件	2010-04-02 11:08

3、(老版本需要导入)

spring-framework-3.0.2.RELEASE-dependencies.zip\org.apache.log4j\com.springsource.org.apache.log4j\1.2.15 - 解包大小为 171.1 MB				
名称	压缩前	压缩后	类型	修改日期
.. (上级目录)			文件夹	
com.springsource.org.apache.log4j-1.2.15.jar	387.4 KB	349.4 KB	JAR 文件	2010-04-02 11:08

2、创建一个POJO对象

```

1.  public class Users {
2.         private Integer id;
3.         private String name;
4.         //对应的 get set 方法
5.     }

```

3、创建一个配置文件

- 1、文件名称与创建的目的地任意(建议在SRC目录下创建applicationContext.xml)
- 2、并导入约束

```

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.         xsi:schemaLocation="http://www.springframework.org/schema/beans
5.         http://www.springframework.org/schema/beans/spring-beans.xsd">
6.  </beans>

```

4、注册对象到容器

```

1.  <!--将对象交给Spring容器管理 -->
2.  <bean name="users" class="com.spring.bean.Users">
3.
4.  </bean>

```

5、测试

```

1.  @org.junit.Test
2.      public void getVoid(){
3.          //1.创建容器对象

```

```

4.      ClassPathXmlApplicationContext classPath= new
        ClassPathXmlApplicationContext("applicationContext.xml");
5.      //2.找容器要对象
6.      Users bean = (Users)classPath.getBean("users");
7.      //3.打印对象
8.      System.out.println("bean:"+bean);
9.      }

```

在创建spring容器是，可通过构造器参数传入多个文件路径

```

1.      ApplicationContext ac=
2.          new ClassPathXmlApplicationContext(
3.              "/spring-web.xml"
4.              ,"/spring-dao.xml"
5.              ,"/service/spring-service.xml");

```

```

<terminated> Test.getVoid [JUnit] D:\JDK\jdk-8u101-windows-x64\jdk\bin\javaw.exe (2018年8月13
log4j:WARN No appenders could be found for logger
log4j:WARN Please initialize the log4j system properly
bean:com.spring.bean.Users@4923ab24

```

二：集成log4j

1、添加log4j依赖

log4j-1.2.17.jar

2、在SRC目录下创建log4j.properties文件

1、并导入配置信息

```
1.  # Global logging configuration
2.  log4j.rootLogger=DEBUG, stdout
3.  # Console output...
4.  log4j.appender.stdout=org.apache.log4j.ConsoleAppender
5.  log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
6.  log4j.appender.stdout.layout.ConversionPattern=%5p [%t] - %m%n
```