PROJECT ONETIME SETUP DOCUMENT

-by RAGHU Sir

# Software Download Links:

**JDK 8 Download: https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads- 2133151.html**

**STS Download Link: https://download.springsource.com/release/STS/3.9.7.RELEASE/dist/e4.10/spring-tool-suite- 3.9.7.RELEASE-e4.10.0-win32-x86\_64.zip**

**Tomcat Download Link:**

**http://mirrors.estointernet.in/apache/tomcat/tomcat-8/v8.5.39/bin/apache-tomcat- 8.5.39.exe**

**MySQL Database : http://www.mediafire.com/file/98auro9hg2f1fwl/MySQLWithYog.rar/file**

**pom.xml:-**

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.1.7.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>5.1.7.RELEASE</version>

</dependency>

<dependency>

<groupId>jstl</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.3.10.Final</version>

</dependency>

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-dbcp2</artifactId>

<version>2.2.0</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.46</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>2.4</version>

<configuration>

<failOnMissingWebXml>false</failOnMissingWebXml>

</configuration>

</plugin>

</plugins>

</build>

AppConfig.java

**package** com.app.config;

**import** java.util.Properties;

**import** org.apache.commons.dbcp2.BasicDataSource;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.ComponentScan;

**import** org.springframework.context.annotation.Configuration;

**import** org.springframework.context.annotation.PropertySource;

**import** org.springframework.core.env.Environment;

**import** org.springframework.orm.hibernate5.HibernateTemplate;

**import** org.springframework.orm.hibernate5.HibernateTransactionManager;

**import** org.springframework.orm.hibernate5.LocalSessionFactoryBean;

**import** org.springframework.transaction.annotation.EnableTransactionManagement;

**import** org.springframework.web.servlet.config.annotation.EnableWebMvc;

**import** org.springframework.web.servlet.view.InternalResourceViewResolver;

@Configuration

@ComponentScan("com.app")

@PropertySource("classpath:app.properties")

@EnableWebMvc

@EnableTransactionManagement

**public** **class** AppConfig {

@Autowired

**private** Environment env;

//1. DataSource

@Bean

**public** BasicDataSource ds() {

BasicDataSource d=**new** BasicDataSource();

d.setDriverClassName(env.getProperty("spring.jdbc.dc"));

d.setUrl(env.getProperty("spring.jdbc.url"));

d.setUsername(env.getProperty("spring.jdbc.un"));

d.setPassword(env.getProperty("spring.jdbc.pwd"));

**return** d;

}

//2. SessionFactory

@Bean

**public** LocalSessionFactoryBean sf() {

LocalSessionFactoryBean s=**new** LocalSessionFactoryBean();

s.setDataSource(ds());

s.setHibernateProperties(props());

s.setAnnotatedClasses(**null**);

**return** s;

}

@Bean

**public** Properties props() {

Properties p=**new** Properties();

p.put("hibernate.dialect",env.getProperty("spring.orm.dialect"));

p.put("hibernate.show\_sql",env.getProperty("spring.orm.showsql"));

p.put("hibernate.format\_sql",env.getProperty("spring.orm.fmtsql"));

p.put("hibernate.hbm2ddl.auto",env.getProperty("spring.orm.ddlauto"));

**return** p;

}

//3. HibernateTemplate

@Bean

**public** HibernateTemplate ht() {

HibernateTemplate h=**new** HibernateTemplate();

h.setSessionFactory(sf().getObject());

**return** h;

}

//4. HibernateTxManager

@Bean

**public** HibernateTransactionManager htm() {

HibernateTransactionManager htm=**new** HibernateTransactionManager();

htm.setSessionFactory(sf().getObject());

**return** htm;

}

//5. ViewResolver

@Bean

**public** InternalResourceViewResolver ivr() {

InternalResourceViewResolver v=**new** InternalResourceViewResolver();

v.setPrefix(env.getProperty("spring.mvc.prefix"));

v.setSuffix(env.getProperty("spring.mvc.suffix"));

**return** v;

}

}

AppInit.java

**package** com.app.config;

**import** org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;

**public** **class** AppInit **extends** AbstractAnnotationConfigDispatcherServletInitializer{

@Override

**protected** Class<?>[] getRootConfigClasses() {

**return** **new** Class[] {AppConfig.**class**};

}

@Override

**protected** Class<?>[] getServletConfigClasses() {

**return** **null**;

}

@Override

**protected** String[] getServletMappings() {

**return** **new** String[] {"/"};

}

}

app.properties

#Jdbc Properties#

spring.jdbc.dc=com.mysql.jdbc.Driver

spring.jdbc.url=jdbc:mysql://localhost:3306/warehouse

spring.jdbc.un=root

spring.jdbc.pwd=root

#ORM Properties#

spring.orm.dialect=org.hibernate.dialect.MySQL55Dialect

spring.orm.showsql=true

spring.orm.fmtsql=true

spring.orm.ddlauto=create

# MVC Properties#

spring.mvc.prefix=/WEB-INF/views/

spring.mvc.suffix=.jsp

#Email Properties#